International Journal of Advanced Science and Technology
Vol. 29, No.02, (2020), pp. 1669-1684

Route Optimization with Guaranteed Fault-Tolerance in IoT using Ant Inspired Hierarchical LCA

Shreyas J∗, Ravi Teja V∗, Sagar C S∗, Srinidhi N ∗, Dharamendra Chouhan ∗, Sunitha G P †, Dilip Kumar S M,*

Dept. of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore, India∗
Dept. of Master of Computer Application, Jawaharlal Nehru National College of Engineering, Shimoga, India†

Abstract: *Internet of Things (IoT) refers to the rapidly growing network of connected devices which assembles and interchange data through embedded sensors of different kinds. In order to deliver guaranteed connectivity among these devices, fault tolerant routing has to be considered. In this paper, a routing optimization for IoT is proposed. The proposed work is a modified version of Least Common Ancestor (LCA) algorithm as a tree based Hierarchical Least Common Ancestor (HLCA) to achieve optimal routing with guaranteed fault tolerance. HLCA algorithm enables efficient transfer of data between devices in a decentralized network that leads to better utilization of resources, little dependency on the super node, which is an intermediate node which supports the transfer of data, thereby improves the overall reliability and robustness of the network. The credibility of the algorithm has been evaluated and has been found to yield better results compared with existing algorithms. The results prove enhancement in the reliability and tolerance of the network with as much as 91% efficiency for various topologies.*

Keywords *: Ant Inspired, Fault Tolerance, Internet of Things, Hierarchical Least Common Ancestor (HLCA), Route Optimization*.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) considered the base of the future IoT. IoT consists of divergent composite network prototype where diversity of components are utilized as consumer electronic devices which interact one among them in a composite way [1]. However, this complex interaction needs to be handled carefully and efficiently by the network. Thus, fault tolerant routing has received attention in the recent years [2]. Therefore, it is required to pay importance to the concept of fault tolerance in routing of IoT. Many works have been carried out in this area of research in the past to make the network more reliable, robust, efficient and fault-tolerant. Often WSNs operate in an autonomous mode without a human supervision in the loop [3]. Moreover, sensor nodes are often deployed in uncontrolled and sometimes even hostile environments [4]. Therefore, it is difficult to predict the optimal way to treat fault tolerance within a particular WSN routing approach, since both technology and applications for IoTs are changing at a rapid pace ,the available communication energy is lower than the computation energy. Therefore, it is important to develop fault tolerance routing algorithms to recover the network from path failure. Consider a network that consist of a large set of super nodes and sensor nodes, super nodes and sensor nodes are interconnected each other. As the number of super node increases the network size also increases, this results in larger distance of hop count and increased delay. In any centralized network the problems like increased hop count, delay and queue length are quite common.

Since the network size is directly proportional to hop count, delay and queue length the network affecting factors also increases with increased network size. In these kinds of centralized IoT network if any sensor node is disconnected, then it has to search for other super node to connect and transfer the data. If no super nodes are present nearby, then fault tolerance is difficult to handle. There will be various challenges that need to be addressed.

- There should be a proper congestion control mechanism to avoid queuing of data in super node.
- Network should continue to work even when some part of the network is down. Therefore, the network should be fault tolerant and continue to offer reliable services.
- For larger amount of data, the network should try to ensure faster transfer of data between sensor nodes.
- Dynamic routing need to be handled to overcome the fault tolerance.

This paper overcomes the various challenges posed, which includes:

- The network construction and selection mechanism is used to construct the network. If the super node is not connected to any node, sensor node gets connected to super node then it becomes a part of one network. Once the super node is created it cannot be removed from the network until all the sensor nodes get disconnected from super node.
- Sensor nodes and super node are connected in the form of tree, where every super node can have maximum of three nodes connected to it.
- There are large number of super nodes in the network to cater the need of large number of sensor nodes which reduces the congestion in the network as the network continues to grow.

On the failure of a super node, the nodes which are connected to the failed super node will disconnected from it and gets connected to another super node to continue the data transfer and to avoid the fault tolerance. The routing of the network is achieved using a modified version of Least Common Ancestor called Ant inspired Hierarchical Least Common Ancestor (HLCA) algorithm.

The contributions of this paper are as follows:

- A network construction and selection mechanism algorithm is proposed to build a network of super nodes with sensor nodes.
- The routing of data from source to destination is carried out by using HLCA algorithm.
- The performance of proposed Ant inspired HLCA algorithms is compared against existing optimization algorithms namely the particle multi-swarm optimization (PMSO), canonical particle swarm optimization (CPSO) and Fully Particle Multi Swarm Optimization (FPMSO).

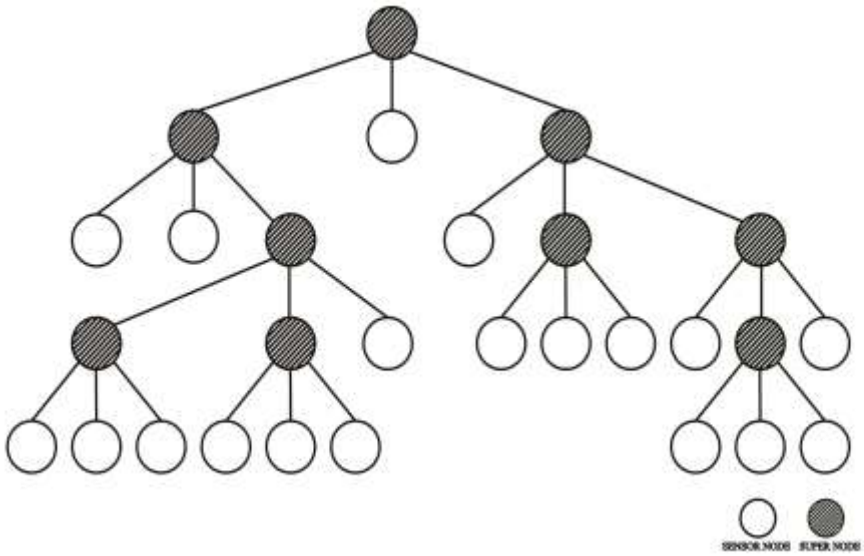
The remaining sections of this paper is discussed as follows: Section II gives the literature survey. Section III gives the problem statement and objectives. Section IV presents the system model. In section V, The proposed construction algorithm and Ant inspired HLCA is presented. Section VI discuss case studies of different scenarios, experimental setup and validates the performance analysis. Section VII concludes the paper, respectively.

II. RELATED WORK

The fault tolerance in routing is generally used to ensure consequent in-dependency, reliability and availability in the network [5]. Furthermore since wireless sensor network devices are connected to the environment with several QoS requirements for tracking and monitoring vital events [6]. The current investigations involve various type of optimization to IoT such as, to improve the efficiency in emergency response and reliable data transmission in IoT [7], introduces a global information decision based routing protocol for emergency response of IoT. The introduced approach utilizes Delay Iterative Method (DIM) to overcome the issues in ignoring valid paths. Furthermore, residual energy probability choice is presented for load balancing. The authors claims the considerable improvement in energy consumption, packet loss ratio and end to end delay but fails to defend for large scale IoT environment.

The authors in [8] investigates Trust Connectivity Probability (T-CP) based optimal routing path to improve multi hop routing performance. The proposed approach considers both fixed and random locations for base stations(BS) based on Connectivity Probability (CP) to obtain the shortest path. Though the algorithm provides the better optimal path but does not satisfactorily improve the overall performance.

The work in [9] proposed security on boarding and routing mechanism for Named Data Network (NDN) of IoT environment. The proposed work adopts hierarchical network design to obtain scalability. The results evaluated in Network simulator-3 proves to be efficient in terms of network overhead but decline to withstand for large set of IoT network. To overcome the IPV6 address issues of



Low power Lossy Network (LLN). The author in [10] proposed diverse bidirectional traffic delivery (DT-RPL) based routing technique.The proposed approach updates the quality of the link by downward and upward traffic to support various traffic patterns. The author claims improved performance in radio energy consumption control overhead and packet delivery ratio but fails to defend his work for various networks of IoT systems. To address the source location privacy problem in routing of IoT system [11], introduces dynamic routing protocol to increase the paths for data transmission. The proposed scheme includes directed routing, greedy routing and cyclic routing to enhance the network security. The proposed work proves consistent battery lifetime. The work in [12], introduces a energy aware routing mechanism for throughput optimization in Clustered WSN's–IoT environment. The proposed approach minimizes the packet overhead, lessens the node level computation and decreases the broadcasting of packets. The proposed technique proves the improved network lifetime but fails to withstand for larger set of nodes.The author of [13], presents a unique mobile matrix routing protocol for mobility management and any to any routing mechanism for IPV6 address allocation. The proposed technique presents mobile cyber-physical network to provide transparency in the device mobility of top layers of network stack. Even though author claims the improved energy efficiency in routing, but the author declined to address different network scenarios. In order to address Rank and Sybil attacks of IoT network and to provide better routing environment, [14] proposed a Sec Trust-RPL routing protocol. The proposed Sec Trust RPl protocol optimizes the network performance by isolating and detecting mechanism based on trust based technique. The proposed approach proves better routing performance by detecting suspicious node. The algorithm fails to withstand for various different kinds of attacks which will greatly effect the network performance. The work in [15], presents a routing technique based on dynamic programming to obtain better path with reasonable resource consumption and great fault-tolerance. The presented approach utilizes Athena Routing Mechanism (ARM) which adopts path probing scheme and source routing to validate the capacities and connectivity of path. The ARM seems to be satisfactory in terms of fault tolerance but restricted to a particular environment. The work in [16], investigates the optimizing multipath routing with guaranteed fault tolerance. The proposed work uses the Particle Multi Swarm Optimization(PMSO) technique to guarantee the connectivity among IoT connected objects. Further while satisfying QoS requirement, the proposed technique selects K-disjoint paths to tolerate the failure. Since the proposed work uses centralized network, it is difficult to transfer the data from node to node because of load shedding and queuing delay.

III. PROBLEM STATEMENT

The problem considered in the proposed work is to find an optimized route for a homogeneous decentralized IoT network to provide faster data transmission with guaranteed fault tolerance.

The following objectives need to be met:

1 Reduce the re-transmission of data.
2 To decrease the delay while transmitting data.
3 Minimizing the queuing delay of data in super nodes.

IV. SYSTEM MODELS

The proposed model considers one-to-many traffic pattern where many sensor nodes are connected to one super node which together guarantees the required connectivity in the network.

Theorem 1 For every tree with N super nodes there are maximum of 2N+1 sensor nodes and minimum of N sensor nodes. A super node contains a maximum of three nodes that can be either super or sensor node, out of which the number of super nodes shall not exceed two.

Proof: For one super node there can be a maximum of three sensor nodes. For two super nodes their can be maximum of five sensor nodes, for three super nodes seven sensors, so on it follows an arithmetic progression.

Number of super nodes 1, 2, 3, 4, 5, 6, 7, 8.....
Number of sensor nodes 3, 5, 7, 9, 11, 13, 15.....

Let a be the starting element of the series and d be the difference between two consecutive elements of the series in which a=3 and $d=T_n-T_{n-1}$. For instance, $d=T_2-T_1=5-3=2$. For n super nodes there will be Tn total nodes.

$$T_n = a + (n-1)d$$
$$T_n = 3 + (n-1)2$$
$$T_n = 3 + 2n-2$$
$$T_n = 2n + 1.$$

If super node is connected to only one sensor node, there will be N sensor nodes to N super nodes. Therefore, minimum of N sensor nodes and maximum of 2N+1 sensor nodes for every N super nodes.

Fig. 1. A simple IoT network tree with super and sensor nodes.

A. Fault Tolerance (FT) Model

As the network is constructed in the form of tree, the fault in the network occurs with the disconnection of network or loss of connection between the nodes. Even after fault arise in the network, the proposed approach maintains continued data transfer. If the destination node does not exist in the disjoint network, the data packet is retrieved in the node till the time of Time to Live (TTL). Whenever the nodes reconfigures to connection state, the data is transmitted over the new path after getting the updated routing information from HLCA. Fault tolerance is to reconfigure the disconnected network to the previous state of connection for continuous data transfer. Fault tolerance is improved as follows:

- Whenever a super node fails, the network is divided into two forests, the transmission between nodes in the forests will still continue to operate normally.
- Whenever the network is divided into separate forests, data required to be transmitted from one forest to another forest will be queued in super nodes for the purpose of avoiding re-transmission of data.
- After failure in the connection of network, devices will reconfigure to reconnect themselves in the network. Once the separate forests are combined together, the super nodes starts transferring the data between the previously separated forests. This leads to the overloaded data transferring at the joined super node. Hence, the collision increases. By using collision detection and control techniques the collision will be avoided.

B. Network Model

Every node (super or sensor) is connected to a super node. The form of connection is a tree T (V, E) where $V=\{v_1, v_2,...v_n, v_{n+1},....v_{N+M}\}$ is a set of finite number of nodes in the network. N and M represents the number of super nodes and sensor nodes respectively. By using Theorem 1, when N super nodes are present, then their exist maximum of M = 2N + 1 sensor nodes. Therefore, the total number of nodes combined is equal to (2N + 1) + N = M + N which is shown in Fig 1.

An edge provides a bidirectional communication channel between a pair of nodes. An edge is required between every pair of nodes, so the total number of edges is one less than the number of nodes in the graph. Therefore, the set of edges E = { e_1, e_2, e_3, ...e_{N+M-1}} provide a bidirectional connectivity in the network. The graph T(V, E) is acyclic graph since the number of edges is one less than the number of vertices and it can be proved that by adding one more edge the graph becomes cyclic.

C. Delay Model

To calculate the delay, transmission delay, queuing in buffer, propagation delay, processing delay, idle time and re-transmission delays are considered. Since a super node is responsible for transmitting data to atmost three nodes connected to it, queuing delay will be less. As a result, the packets get transmitted quickly from one super node to another super node and reaches the sink. The delay from source to destination is computed as follows:

$$D_{s,d} = D_{que} + D_{prop} + D_{proc} + D_{trans} + D_{retran} + D_{load} + D_{idle} + D_{ack} \quad (1)$$

where, D_{que} is queuing delay at super node, D_{prop} is time taken to arrive at destination, D_{proc} is time taken to prepare and process (packing from layers) the packet, D_{trans} is transmission delay, time taken to transfer a packet, D_{retran} is time taken to transfer lost packet again to destination, D_{load} is loading delay, D_{idle} is delay of router during the idle state of system. The total delay taken while transferring the packet from source s to destination d i.e ϕ_{tot} can be computed as follows:

$$\phi_{tot} = \sum \phi(i,j) \leq \Delta\phi \quad (2)$$

where $\phi(i,j)$ be the number of nodes present in the path between source to destination, $\Delta\phi$ is the bounded delay. The delay from source node to the LCA and the delay from LCA to the destination node.

V. PROPOSED ALGORITHM

A. Network Construction and Node Selection Algorithm:For construction of the network, a construction and node selection algorithm is proposed. The algorithm selects a sensor node x and finds a super node to which it can connect. If no such super nodes are found, then a super node y is created and the sensor node x is connected to it. The newly created super node y joins the network by replacing one of the sensor node x, with the new super node y. Once the network construction is completed, the selection of the path is computed as follows:

- Consider any two nodes A and B within the coverage distance C_d
- The algorithm selects a super node y and connects A and B to it.
- The nodes previously connected to super node y is replaced within the coverage distance C_d

The network construction and selection mechanism algorithm is presented in Algorithm 1.

Algorithm 1 Construction and Selection Mechanism Algorithm

```
procedure
    INPUT SET of M SENSOR NODE
   for each i ∈ M SENSOR NODE/(S) do
     if (Any super node is Ø) &&
         ( Dist(supernode,i) < Cd ) then
         Connect i → Supernode
     else
         Create a Super node in Cd.
         if (Removal of Sensor node X is Successful)
            && (Count of connections
            to super node≤2)) then
            Connect Created Supernode→Supernode Y
            Connect Sensornode X →
            created Supernode
```

```
            Connect i → Created Supernode
        else
            Choose another Super node, go to Step 9
        end if
    end if
end for
NETWORK = Constructed Network;
for (i,j ∈ NETWORK) && ((i,j)NotV isited) do
    if i ≠ j && distance(i, j) ≤ Cd then
        Choose Super node K
        Connect i → K in place of c
        Connect j → K in place of d
        Connect d → Previous i's Super node
        Connect c −→ Previous j's Super node
        Mark (i,j) as Visited
    end if
end for
end procedure
```

Algorithm 2 Ant Inspired-HLCA

```
procedure HLCA
    X=Source (S), Route={Source}
    Y=Destination (D)
    while (X! = Y ) do
        P=Set of neighbour node of x
        for Each m in set p do
            if RSSI (S, m)+RSSI (D, m) is greater then
                X = M
                route=route+X
            end if
        end for
    end while
end procedure
```

B. Hierarchical Least Common Ancestor (HLCA)

LCA algorithm is used to find the distance between two nodes and finding the common node between any two nodes [17]. LCA algorithm runs every time to find path from source to destination for routing the data. The proposed HLCA algorithm finds the path and stores information in the routing table for continuous data transfer. In case of failure in path, the HLCA algorithm finds and updates the changes in routing table. If A wants to send data to B, A forwards the packet to its ancestors until it finds the least common ancestor that has B as its descendant. The HLCA maintains and forwards the packet down the subtree. Failure to find any common ancestor with such information will result in updated routing table by running the HLCA algorithm. The proposed HLCA algorithm in finding optimal path between any two nodes is presented in Algorithm 2. The algorithm considers a node R as root of the network and two nodes A and B. The path for transmission between A and B to which the path is to be computed. Let the path P_A = Ra1,a2,a3,....A, represents the path from root R to A, the path P_B= Rb1,b2,b3,...B represent the path from root R to B. The common node L in path P_A and P_B is the Least Common Ancestor of the nodes A and b. Let P'_A the path from node L to A and P'_B be the path from node L to B. The concatenation of the paths P'_A and P'_B is the path for transmission between A and B.

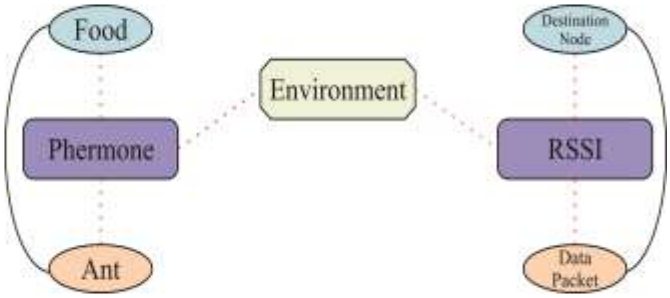


Fig. 2. ACO and ACO-inspired HLCA

C. Mapping of ACO to HLCA

Inspired by the real ants, the Ant Colony Optimization (ACO) algorithm is a kind of probabilistic technique used for solving computational problems. In this section, ACO is mapped to HLCA with respect to routing environment, the mapping illustrates the benefits of ACO-inspired HLCA. The proposed ACO-inspired HLCA frame work is also explained.

- Environment Constraint: As shown in Fig. 2 ACO consists of three important roles. Namely, pheromone, food and ant. The ants are considered as data packets and food is considered as destination node. The pheromone plays an important role in connecting ant with food. In the proposed approach Received Signal Strength Indicator (RSSI) is used to determine the neighboring node hence, RSSI is regarded as pheromone. RSSI is the amount of signal received from node in response to the request message. In the proposed approach RSSI is used to determine the neighboring node to establish path between source and destination. The illustration of ACO and ACO-inspired HLCA is presented below:

1) The intention of ACO is to locate food, and swarm of ants travels based on pheromone in a communicational and cooperative way; Based on RSSI value, the nodes communicate and cooperate to send data packet to the destination.

2) The natural behavior of ant makes pheromone decrease or even vanish as time expires. In the same way, the RSSI decreases and even becomes null as the distance between node and destination increases.

3) During the food searching phase each ant lays the pheromone and every ant communicates among each other indirectly based on pheromone concentration; In the proposed method, the data packet determines RSSI value of destination node to the path and nodes communicates each other based on RSSI value. As a, result neighboring node is selected based on RSSI value.

Routing Scenario: The proposed work considers the ant foraging technique as a routing technique. The data packet determines node's RSSI value from network. The location in ant colony optimization consist of three components i.e. food warehouse to gather food, pheromone matrix to precept pheromone, and Tabu Search table to guide travel direction of the ant.

Same way in the proposed method, each node in the network consist of three components, i.e. source data packet, which is the information need to be transmitted. RSSI value is to find the neighboring node, HLCA routes are the paths stored in the routing table. The source data packet, RSSI Value and HLCA routes are correspond to food warehouse, pheromone matrix and tabu search table of the standard ACO algorithm. The routing scenario of ACO and ACO-inspired HLCA is shown in Fig 3. The ant forging technique is described as follows. Each ant travels from nest in search of food. When the ant arrives at the food location, it searches If the ant finds the food then ant completes foraging process and returns back to nest. If food is not found then, ant precepts pheromone by pheromone matrix and travels according to tabu search table. Similarly, the ant behaviors are mapped into data packets and each data packets determines the RSSI value of the node. It checks for increased RSSI value of destination node. Path having better RSSI value is considered as a neighboring node. Data transfer between source to destination happens according to HLCA method.

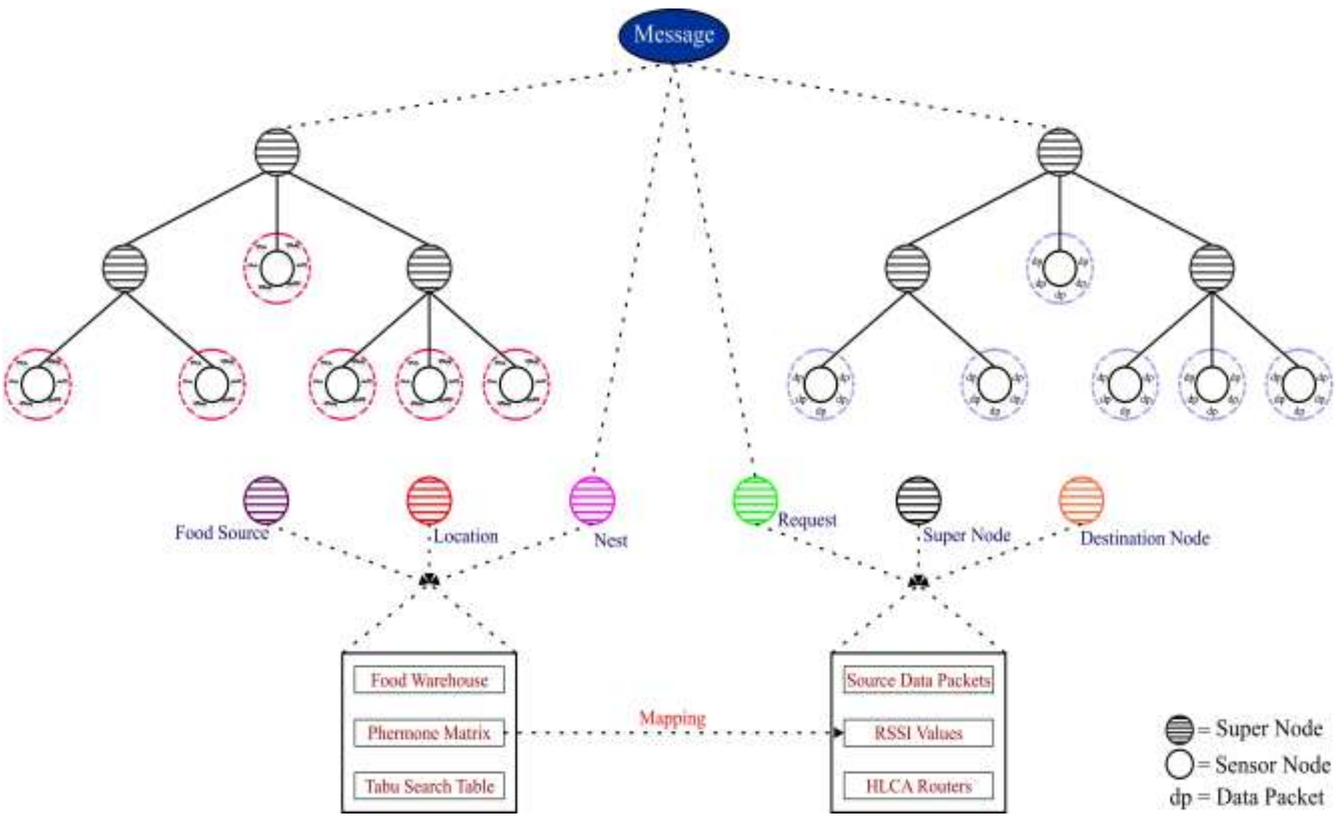


Fig. 3. Routing Scenarios of ACO and ACO-inspired HLCA

[D. Route optimization with proposed network

Generally in WSN, for the transmission of data from source to destination binary tree structured tree is used. In the proposed method a tree structure in which root with maximum of three children is used for data transmission. A binary tree can accommodate 2^{l-1} nodes in a tree of height l. In the proposed topology it can accommodate $3.2^{l}-2$ nodes. In order to prove the above statement, mathematical induction is used. $1 + 3 + 6 + 12 + ... + 3.2^{l}-1 = 3.2^{l} -2$

Basic Step: By considering l=1

$3.2^{0-1} + 3.2^{(l)-1} = 3.2^{(l)-1}$ (3)

$3.2^{-1} + 3.2^{0} = 3.2-2$ (4)

$3/2+3=6-2$ (5)

$1.5 + 3$ (6)

$1 + 3 = 4$ (7)

$4 = 4$ (8)

Hence, LHS = RHS (9)

Assumption Step: We assume that it is true for l = k

$1 + 3 + 6 + 12 + ... + 3.2^{k-1} = 3.2^{k} -2$ (10)

Induction Step: We need to prove for l = k + 1

$1 + 3 + 6 + 12 + ... + 3.2^{k-1} + 3.2^{k} = 3.2^{k} -2+ 3.2^{k}$ (11)

$= 3.2^{k}(1 + 1)-2$ (12)

$= 3.2^{k}.2-2$ (13)

$= 3.2^{k+1} -2.$ (14)

Hence, by mathematical induction it is proved. Fig. 4 shows total number node at level L for 3.2^{L-1} and 3.2^{L-2}.

The height of the proposed topology is smaller than binary tree structure for given number of nodes. As the number of nodes increases the levels in binary tree also increases compared to proposed tree, which is shown in Table I. Hence the number of levels is optimized in the proposed method

TABLE I

NODE AND LEVELS COMPARISON OF BINARY TREE AND PROPOSED TREE

Nodes	**Levels in Binary Tree**	**Levels in Proposed Tree**
100	7	6
500	9	8
1000	10	9
10000	14	12
10^5	17	15

For a given number of nodes and levels, the optimal number of children is to be determined. Let N be the number of nodes, l be the number of levels in a tree and m be the number of children for a parent. The number of children a node can have (i.e relationship between N, m and l) is given by

$$N = m^l - 1 \qquad (15)$$

R /√2 is experimentally considered as the range where any two node can communicate. In the proposed method, as shown in Fig. 5, the optimal number of children is chosen as 3 because only 3 child nodes can fall within in the communication range of the super node and the value is R /√2 [18].

∀m > 3, the child nodes will not be in the R /√2 range. Hence, For HLCA the optimal number of child nodes is considered as 3

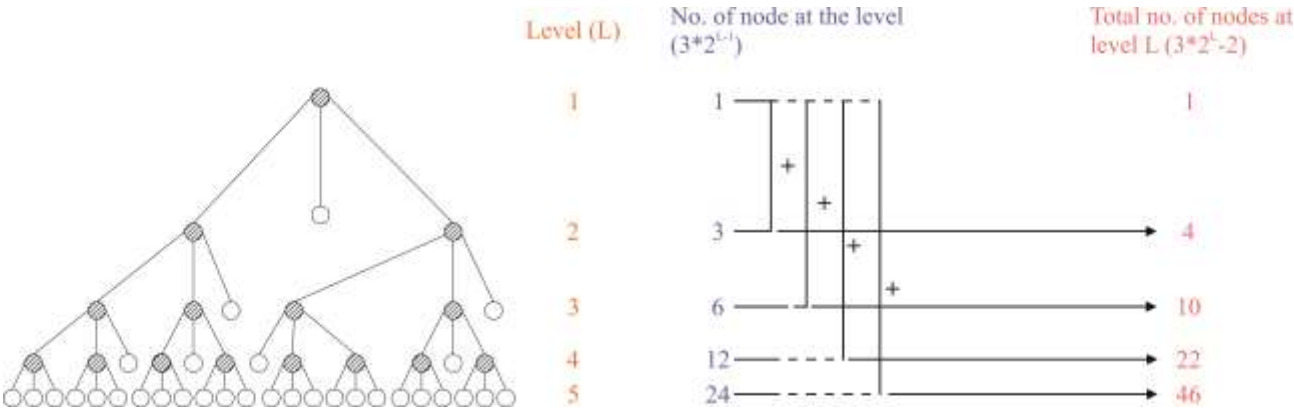


Fig. 4. Number of nodes in Level 4

VI. EXPERIMENTAL SETUP AND PERFORMANCE EVALUATION

A. Network Construction Elements

To validate the effectiveness of the proposed algorithm, a hardware implementation is setup. The hardware components used for implementation are listed below.

1) For the construction of the wireless network Esp8266 (WIFI), proximity sensor, IR sensor, gas sensor, passive infrared sensor, accelerometers sensors, silicon wires, Node MCU[1.0], routers, arduino boards (UNO,MEGA2560) components are used.
2) Routers and Node MCU are used as a supernodes.
3) Arduino UNO, Arduino and Node MCU are used as a sensor nodes.
4) In the network Arduino and Node MCU will be connected with the sensors to route the sensor data to super nodes.

B. Analysis

In order to analyze the proposed algorithm three cases have been considered namely, data rates for low, moderate and high. These three cases are explained briefly in the following.

1) Low rate data transfer: The data transmission between nodes connected to same super node transfers quickly. network topology places the nodes with higher data transmission rate closer to minimize the hop count. The data transmission between two sensor nodes connected to two different super nodes transfers independently with other two sensor nodes connected to different super nodes. Since a super node is responsible for transmitting data to maximum of three nodes connected to it, the queuing delay is less. Since the proposed work places the node within coverage distance, propagation delay is reduced as a result packets reach before TTL and reduces re-transmission.

2) Medium rate of data transfer: Even in case of medium rate of data transfer, the network behaves similar to low rate of data transfer.

3) High rate data transfer: In this case, the network throughput may slightly increase. As the level increases, t he topmost node which has heavy load needs to be transferred. Propagation delay increases as the load increases in the
network [19]. At the leaf level of the tree there will be no overload for super node because it needs to monitor only three nodes, hence there will be no effect on the leaf level of sensor nodes. As the data transmission rate gets increases, the congestion in network arises. Hence, re-transmission delay slightly increases in this scenario.

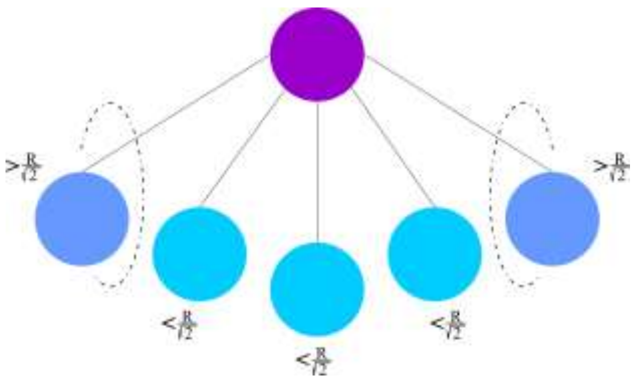


Fig. 5. Communication range of a super node.

C. Performance Evaluation

The performance of the proposed work is evaluated by constructing the network 256 super nodes and 500 sensor nodes. We compare our proposed approach against the CPSO, FPMSO and CPMSO algorithms.In order to compare the performance of the Hierarchical Least Common Ancestor algorithm, the following three performance metrics are used: Delay, Throughput and retransmission.

1) Delay: The delay is calculated evaluated by considering average delay in transferring data packet from source to destination at specific period. The experimental outcome indicates that the proposed Ant inspired HLCA is effective. The X-axis represents the number of iteration and Y-axis is represented as average delay in transferring data packet. Fig. [6-11] shows the average delay in the all three case mentioned as low, medium and high density is comparatively better compared with CPSO, FPMSO and CPMSO algorithms. However, their is a slight increase in delay in high data transfer Fig. 7, 9 and 11 but still results are effective compared to existing protocols. The existing protocol uses a centralized network in which load will be increased at the super node. Load is directly proportional to the propagation delay in the network. In order to avoid the loops in the network the proposed algorithm uses decentralized network hence prorogation delay decreases and the performance of the network increases.

2) Throughput: The second metric, Throughput is evaluated by considering number of Kbs that are successfully transmitted by the source nodes in a network per second. he X-axis represents the number of iteration and Y-axis represents throughput achieved. The performance of the network increases with higher throughput, As the queuing delay decreases, the throughput increases in the proposed approach. Though the one super node is congested all other super nodes are non congested hence, the maximum

throughput is achieved. Fig. [12-17] shows the average throughput in all the three cases mentioned as low, mediumand high density is comparatively better than CPSO, FPMSO and CPMSO algorithms. However, Fig. 13, 15 and 17 shows the slight downfall of throughput in case of high data transfer but still achieves effective results compared to existing protocols.

4)Re-transmission: The third metric re-transmission is calculated as number of packets re-transmitted versus number of iteration. As the delay is directly proportional to retransmission, the re-transmission of data reduces. Fig. [18-23] shows the re-transmission in all the three cases mentioned as low, medium and high density is comparatively better than CPSO, FPMSO and CPMSO algorithms.

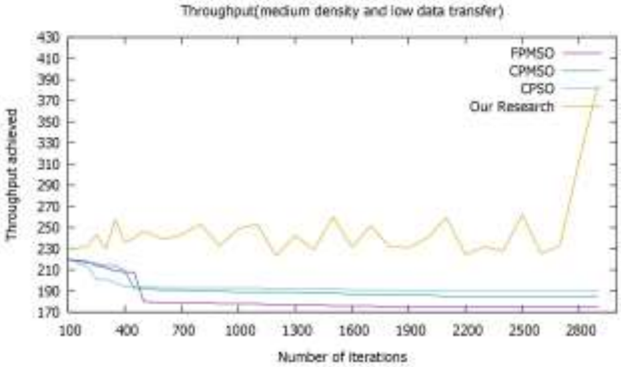


Fig. 6. Throughput comparison of medium density low data transfer

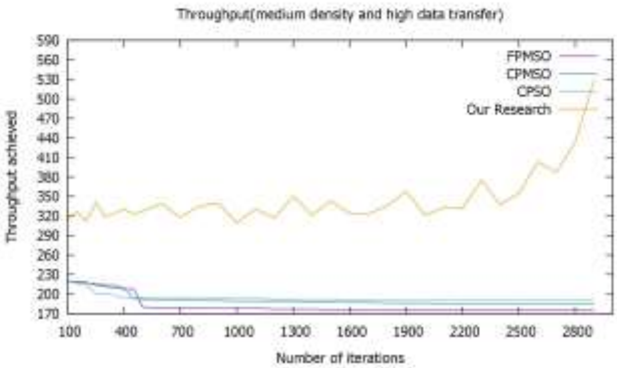
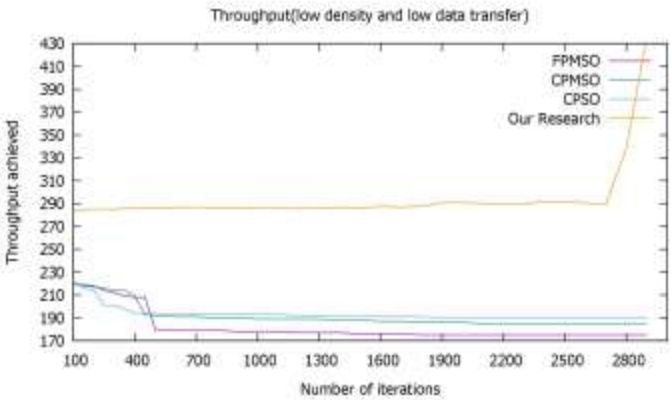
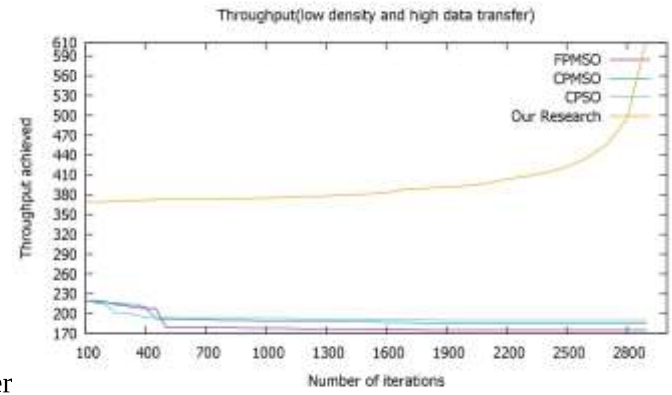


Fig. 7. Fig. 15. Throughput comparison of medium density high data

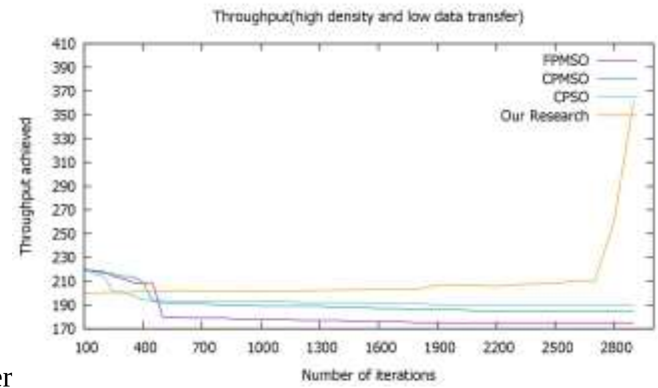


transfer

Fig. 8. Throughput comparison of low density low data



transfer
Fig. 9. Throughput comparison of low density high data



transfer
Fig. 10. Throughput comparison of high density low data transfer

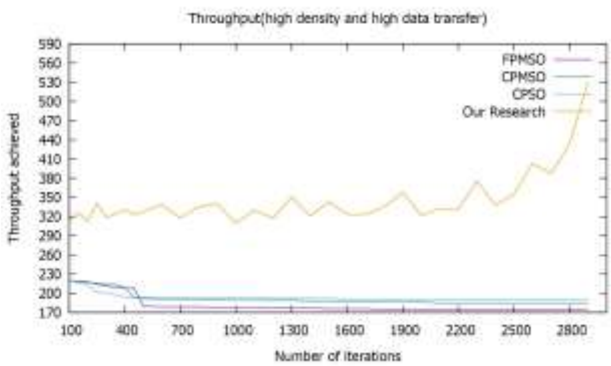
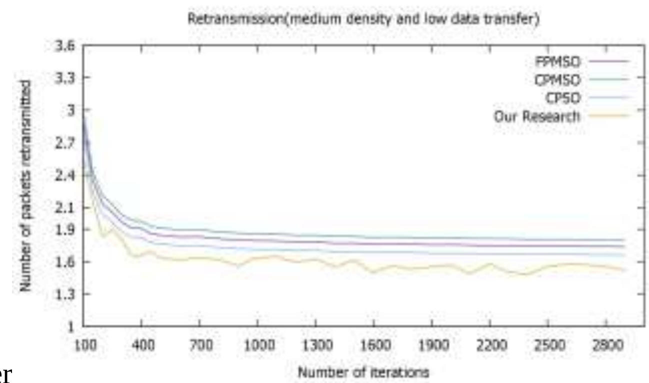


Fig. 11. Throughput comparison of high density high data



transfer

Fig. 12. Re-transmission comparison of medium density low data transfer

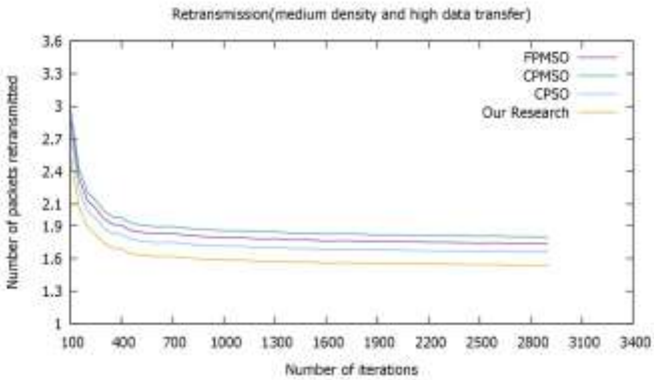
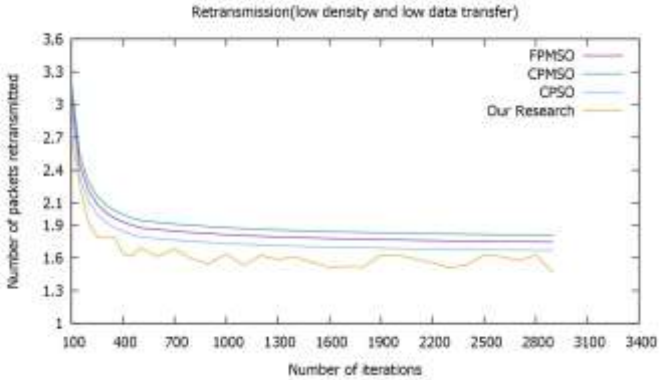


Fig. 13. Re-transmission comparison of medium density high data



transfer

Fig. 14. Re-transmission comparison of low density low data transfer

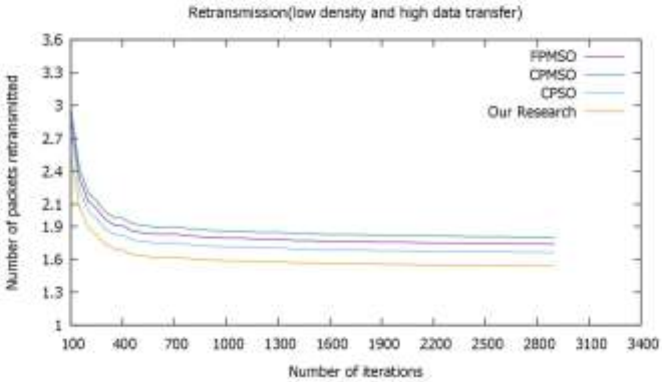


Fig. 15. Re-transmission comparison of low density high data transfer

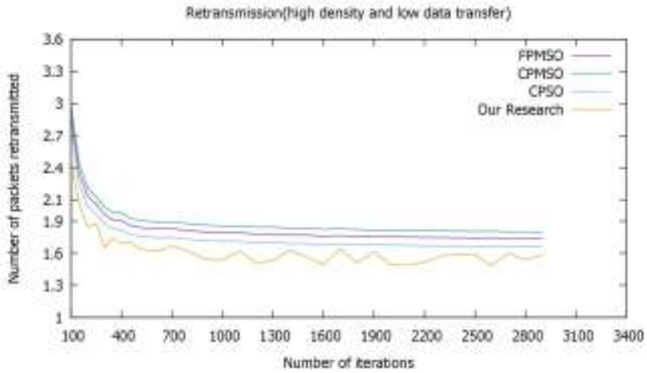
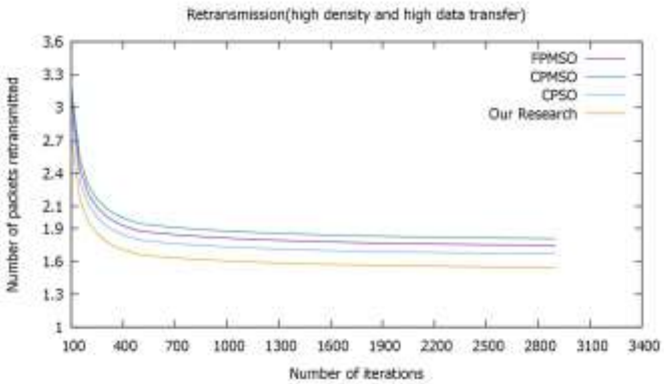


Fig. 16. Re-transmission comparison of low density low data



transfer

Fig. 17. Re-transmission comparison of high density high data transfer

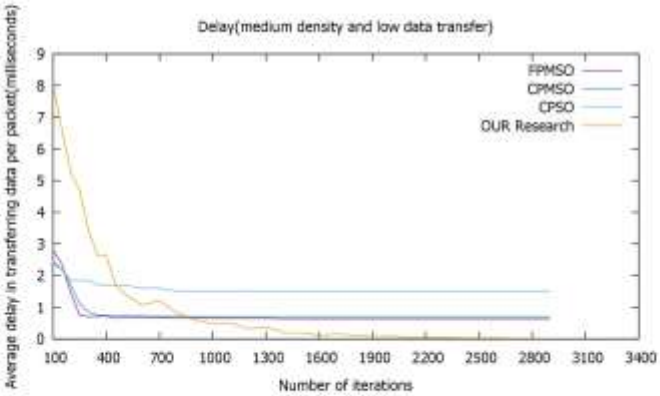


Fig. 18. Average delay comparison of medium density low data transfer

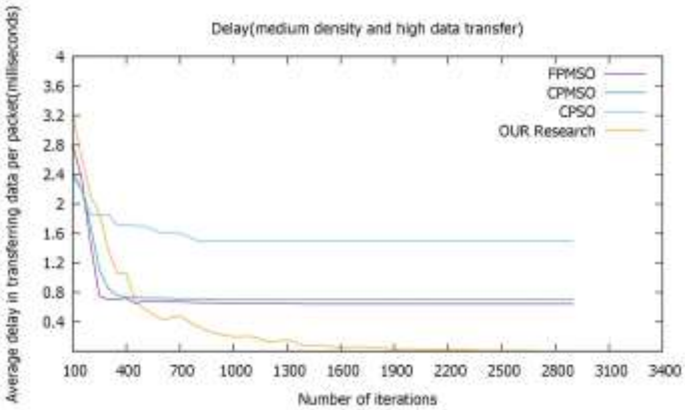
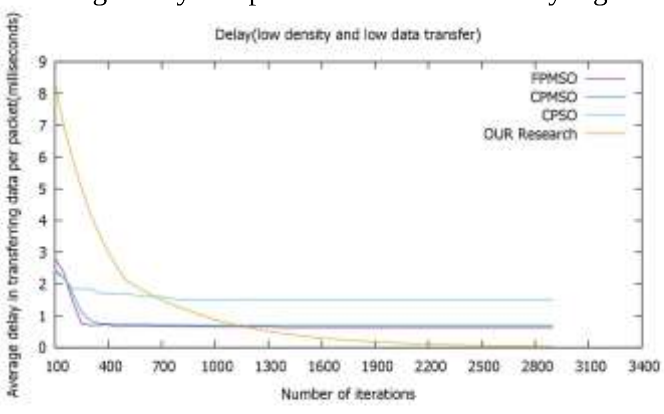


Fig. 19. Average delay comparison of medium density high data



transfer

Fig. 20. Average delay comparison of low density low data

ISSN: 2005-4238 IJAS

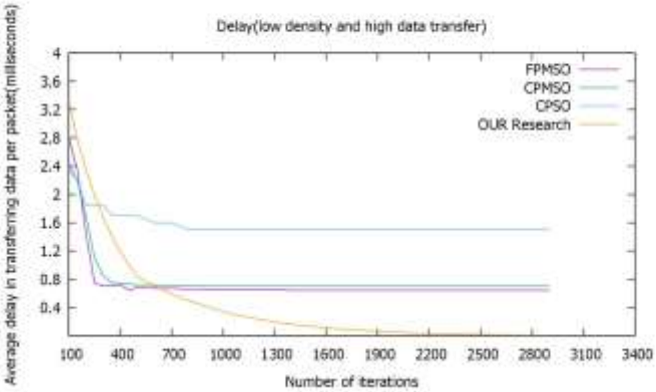


Fig. 21. Average delay comparison of low density high data transfer

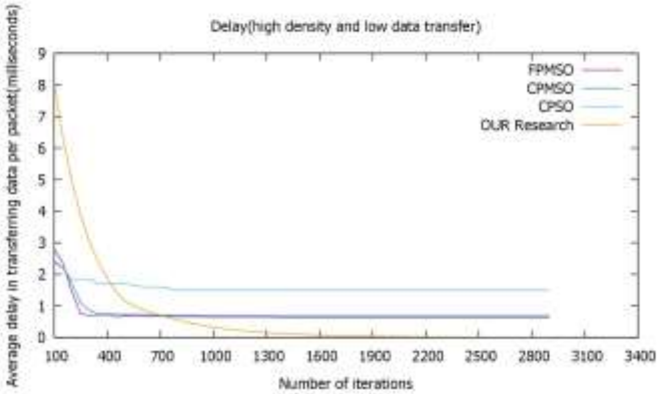


Fig. 22. Average delay comparison of high density low data transfer

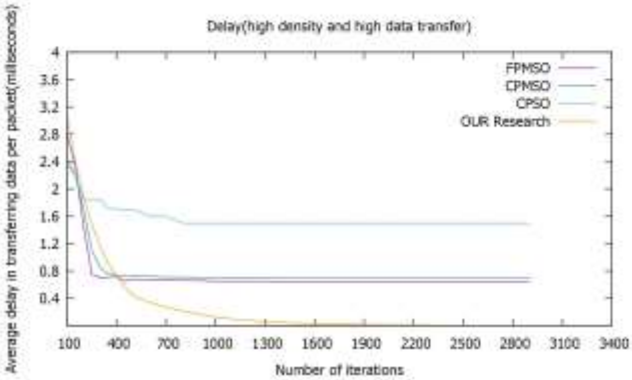


Fig. 23. Average delay comparison of high density high data transfer

VII. CONCLUSION

In this work, an optimal path for routing has been proposed. The proposed Ant inspired Hierarchical Least Common Ancestor Algorithm is proposed to achieve the optimal routing. The Proposed algorithm algorithm speed up the transmission time by reducing re-transmissions and delay in the network path by increasing throughput. Through the proposed network construction and node selection algorithm, the network becomes fault tolerant by adopting strategies to make the network recover back to its functional state. The proposed approach have been compared with existing algorithms to show the advantage of proposed work. The results of the proposed work shows the performance in efficiency of throughput is increased to a 91%.

ACKNOWLEDGMENT

This research work has been funded by the Science and Engineering Research Board (SERB-DST) Project File No: EEQ/2017/000681. Authors sincerely thank SERB-DST for intellectual generosity and research support provided.

REFERENCES

1. F. Al-Turjman and G. Singh, "A data delivery framework for cognitive information-centric sensor networks in smart outdoor monitoring 1," in Cognitive Sensors and IoT. CRC Press, 2017, pp. 107–148.

2. M. Z. Hasan, H. Al-Rizzo, and F. Al-Turjman, "A survey on multipath routing protocols for qos assurances in real-time wireless multimedia sensor networks," IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1424–1456, 2017.

3. Y. Zeng, L. Xu, and Z. Chen, "Fault-tolerant algorithms for connectivity restoration in wireless sensor networks," Sensors, vol. 16, no. 1, p. 3, 2015.

4. N. Srinidhi, S. D. Kumar, and K. Venugopal, "Network optimizations in the internet of things: A review," Engineering Science and Technology, an International Journal, 2018.

5. G. Colucci, M. Poletti, R. Stefanelli, and D. Trinchero, "Internet of things as a means to improve agricultural sustainability," in Biomedical Circuits and Systems Conference (BioCAS), 2017 IEEE. IEEE, 2017, pp. 1–4.