RIVER: A Bio Inspired Routing Protocol For High Data Rate Wireless Sensor Networks

1st Geetha V
Dept of Information Technology
NITK Surathkal
Mangaluru, Karnataka, India
geethav@nitk.edu.in

2nd Priya Malik
Dept of Information Technology
NITK Surathkal
Mangaluru, Karnataka, India
priyanjuma@gmail.com

3rd Sowmya Kamath S
Dept of Information Technology
NITK Surathkal
Mangaluru, Karnataka, India
sowmyakamath@nitk.edu.in

***Abstract*—Wireless sensor networks are deployed to observe the environment such as disaster management facilities and some industrial applications to observe temperature, pressure and humidity etc. The network is dynamic, battery operated and as a result it phases many issues compared to other networks like MANETs. Compared to MANETs, the issue or challenges are more as the nodes in the network has to send data to a common node called sink node. The challenges increases if the data type is other than scalar data. In future WSN gets extended to high data rate applications to sense and transmit multimedia, image or audio files. Normal routing algorithms, which are currently suitable for low data rate WSNs (LWSNs) are not sufficient for HWSNs. As a result, a new type of routing protocol which can solve various issues of HWSN is essential. It has been found that bio inspired algorithms perform very well for optimization and combinatorial problems. Bio-inspired algorithms are inspired from nature and nature has found very good solutions to very complex problems. In this regard, we are proposing and developing a new bio inspired routing protocol for high data rate wireless sensor networks called River which is based on the Intelligent Water Drops algorithm related to swarm intelligent techniques. The proposed method is implemented in ns-2, and the results are compared with AODV. The results shows on an average 75% improvement with respect to various parameters.**

***Index Terms*—NS2, Bio Inspired, Routing Protocol, Ant Colony, Intelligent Water Drop, River**

I. Introduction

Wireless sensor Network (WSN) alludes to a gathering of spatially scattered and devoted sensors for observing and recording the physical states of the earth and sorting out the gathered information at a focal area. WSNs measure ecological conditions like temperature, sound, contamination levels, dampness, wind pace and direction, pressure,etc. A sensor node has four principle units: Sensors to detect physical phenomena, Data processing unit, Wireless transmission interface for communication and Batteries for energy requirements of nodes. Al-Obaisat et al. [1] have given the primary components that develop the architecture which contains sensor node, sink node, phenomenon and the client. Saleem et al. [2] mentions properties which a routing protocol for WSNs must fulfill: Minimal Computational and Memory Requirements, Autonomicity and Self-Organization,Energy Efficiency, and Scalability. In this way, the routing protocol ought to have the capacity to adequately adapt to the difficulties getting from serious radio impedance, long ways, and unpredictable failures. Also, it ought to have the capacity to show adaptable execution in face of these difficulties.

High Data Rate WSN (HWSN) deals with the data of large size eg. an audio, video or image etc. In case of HWSN, information to sense and transfer is huge and is likely to be represented in an array or stream. Zacharias et al. [3] mentions the difficulties to be handled in such scenarios such as quantity of data produced, streaming of data, computation and communication cost, and challenges of directional sensors. To overcome these challenges, the node must be equipped with components allowing them to adjust to the present network status, for neighbor discovery or controlling network congestion. Consequently, bio-inspired methodologies appear to be encouraging since they are exceptionally equipped for self-adjustment, in spite of the fact that they can be at somewhat ease back to adjust to ecological changes. Even though, the use of bio-inspired methodologies in Information Technology is a familiar technique, a large portion of the past endeavors have been focused on optimization issues in control of networks. Further improvements on bio-inspired techniques may provide better results in terms of versatility, flexibility, power and self-association.

Section II describes related work, the proposed bio-inspired routing protocol is explained in section III, and results and discussions are provided in section IV followed by conclusion and future work.

II. Related Work

Saleem et al. [2] have given the taxonomy of routing protocols as:Single-path and Multi-path, Proactive, Reactive and Hybrid, Next hop and Source, Flat and Hierarchical routing, Address and Data centric, Centralised and Distributed,QoS aware and Best effort, Query based and event driven, Energy aware, Loop free, Fault tolerance and Load balancing. A survey by Binitha et al. [4] on bio-inspired algorithms and have given classification for the bio-inspired computing as: 1. Evolutionary Algorithms 2. Swarm Based 3. Ecology Based algorithms.

There are many algorithms which comes under the category of Swarm Intelligence (SI). Some of the examples are Cat swarm optimization, Glow Swarm Optimization (GSO), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony optimization (ABC), Intelligent Water Drops algorithms (IWD) etc. Evolutionary algorithm wants the population details (such as topology) beforehand. It makes whole process quite centralized [5]. Swarm based algorithms are best suitable as they are distributed in nature itself [6], [7] and [11]. Ecology Based Algorithms are quite complex in nature; hence these algorithms are not much suitable for WSNs. Most of the Bio-Inspired routing protocols employed on WSNs are based on Ant Colony algorithms [8], [9], [10], [12], [13], and [16]. Swarm based algorithms [6] and [8] can be explored for use on WSN, such as PSO, GSO, FSA, IWD etc. The algorithms developed for Manets can also be verified for WSNs.

This work focuses on developing a bio inspired Water drop flow model based routing protocol for high data rate wireless sensor networks based on swarm intelligent algorithms.

III. Proposed Bio-inspired routing protocol for high data rate wireless sensor networks

We propose River, as a bio-inspired routing protocol for specific use in high data rate wireless sensor networks. Its working principle is IWD (Intelligent Water Drops). IWD comes under the category of Swarm Intelligence (SI) bio-inspired algorithms, where swarms are considered to be the water drops. IWD tries to mimic the behaviour of the Natural River System. The River flow model is shown in Figure 1.

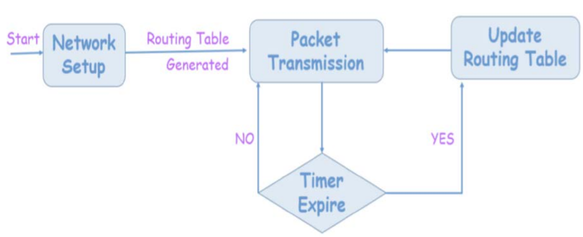

Fig. 1. River Flow Model.

River protocol works in two phases: first phase is network setup phase and the second phase is data transmission phase. During network setup phase, all the nodes are informed about their location and position in the network. At the end of this phase, every node will have information about its neighbors and its hop distance from the sink node. Initial routing table is also created during this phase. The second phase has three functionalities. Packet transmission, expiration of timer and updating routing table. The next hop is calculated from the routing table and the data transmission starts. Whenever a data packet is received at any node, it will deposit some soil on it. After some time, The soil counter will reach its maximum limit (set by the user). This node will then inform all its neighbors about this event of soil timer expire. All the neighbors on receiving this information will update their routing tables by updating the soil value corresponding to the source of the packet received and select a new next hop which has the minimum soil deposited on them. Again data transmission will start and forward the packet to newly selected next hop.

We consider a data packet in network as Intelligent Water Drop. When a packet (IWD) moves from one node to another, it deposits a unit of soil to destination node and the amount of soil carried and deposited by every IWD is also equal.

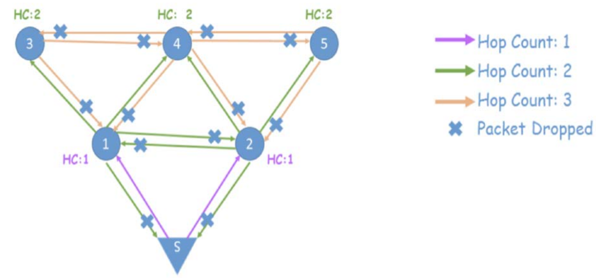

Fig. 2. River Network Setup .

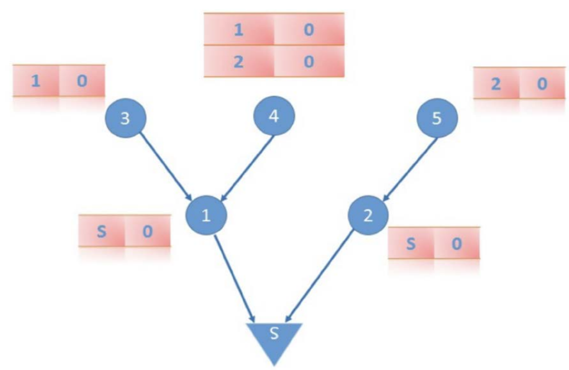

Fig. 3. River Forward.

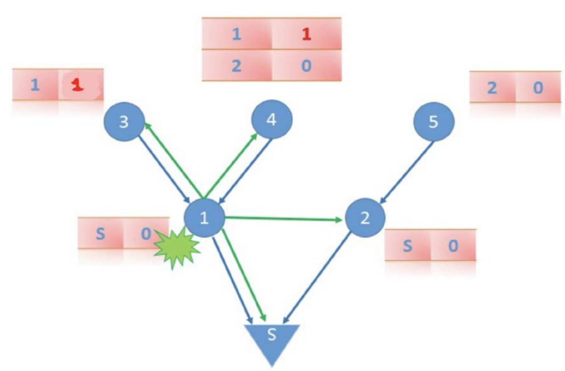

Fig. 4. River Soil Increment.

A. *Initial Network Setup*

The nodes are considered in hierarchical deployment and nodes can be in same level, higher level or low level. The

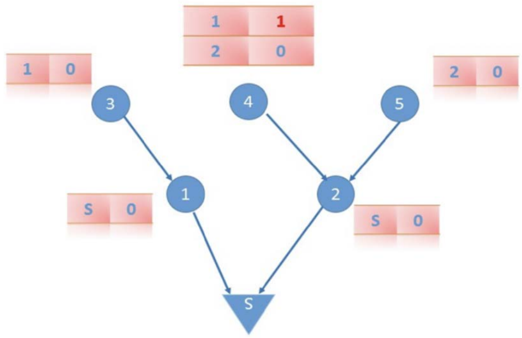

Fig. 5. River Changing the Next Hop

sink node sends network setup packet with a hop count 1 to the neighbor nodes. Neighbor nodes of sink are those nodes which are in its transmitting range. All These nodes will set their own hop count to 1, and then increment the hop count to 2, makes an entry in the routing table and forward it to its neighbors. Now, all the receiving nodes will set their hop count values, increment the hop count in packet, makes an entry in the routing table and forward it to all its neighbors. This continues until all nodes set their hop count values. The process is shown in Figure 2. The sink node broadcasts the network setup packet with hop count 1. Node no. 1 and 2 receives this message. They make an entry to their routing table, set their own hop count as 1, increment the hop count in the packet and again broadcast it. The broadcast message from node 1 is received at node 2,3,4 and S. Node 3 and 4 will perform the same functions as node 1, when it received the network setup packet. Node 2 will drop this packet as it has come from the node in the same level. Node S will also drop the packet, as it is the sink node, and this information is of no use to it. When a node receives a network setup packet, it either accepts this packet or drops it. We enforce that a node should receive network setup packet only from its lower level. A new field is added to routing table called soil and initially the soil value in the routing table is set to 0. Only nodes which are in the lower level, can be added to the routing table.

B. Forwarding Data

Once the routing table is formed and any node receives a packet, it will forward it to the node which has the minimum soil content. Initially, the soil value is set to 0 for all nodes, and in that case, the first entry of routing table is chosen as next hop for packet forwarding. Figure 3 shows the process of data transfer in proposed River routing protocol. Nodes 1,2,3 and 5 are using their only neighbor in the lower level to transfer the packet. But node 4 has two choices, out of which, it has chosen the first entry of the routing table to forward its data packets.

C. Method for Updating Soil Value in Routing Table

Every node keeps a counter for its soil. This soil counter will be incremented with every incoming data packet. When this counter expires a soil increment packet (which contains node id and its hop count) is sent through broadcast. When a node receives this soil increment packet, it increments the soil amount in its routing table by one corresponding to the source node of the received packet. Now, if the soil value incremented was the minimum soil in the routing table, this node will also broadcast soil increment packet further. In this case, the minimum soil value will be decreased from every soil value present in the table. This is to prevent the overflow of soil counter. This packet type is also received only from lower levels to the higher levels. When a node receives a soil increment packet, it takes a decision on whether to accept this packet or drop based on hop count of receiving node is equal to hop count of sending node + 1. In this case the node Accept the packet otherwise it rejects the packet.

Figure 4 shows what happens when a timer expires. It shows that the timer is expired on node 1. This node then broadcasts a timer expire packet. This packet is received by node 2,3,4 and S. Node s drops this packet being the sink node. Node 2 drops this packet being on the same level as the sending node. Node 3 receives receives this packet, increment the soil corresponding to node 1 in its routing table, and finds the minimum soil, which is 1. The minimum soil in the routing table is not 0, in this case, node 3 will broadcast a soil increment packet. After that it subtracts the minimum soil from its only entry in the routing table. As a result the soil value becomes 0. Node 4 receives this packet, increments the soil corresponding to node 1 in its routing table. Finds the minimum soil in routing table, which is 0. Because there is a node in the routing table with soil 0, node 4 will not broadcast the soil increment packet further.

D. Changing next hop

When soil is incremented in the table, a next hop (which has minimum soil) is calculated and the data packets are then forwarded to this new next hop. During soil update process, node 4 incremented the soil corresponding to the node 1 in its routing table. Now it will find the next hop. The node with minimum soil is node 2. Hence, next hop is selected at node 4. Figure 5 shows the data transmission using the new next hop. let us consider that the timer at node 2 also expires after some packet transmissions. Node 2 broadcasts the soil increment packet. Node 1 and s will drop the packet. Node 4 and 5 will accept it. This time, both 4 and 5 will broadcast the soil increment packet. Node 4 does so, because there is no entry with soil 0 in its routing table anymore. After broadcasting the soil increment packet, nodes will find the minimum soil in the routing table and will subtract it from all the soil values in the table. This subtraction is done to prevent the soil values in routing table from overflowing.

The proposed routing protocol provides following features:

- Every node takes its routing decision based on its immediate neighbors. The nodes need not consider global topology.

- Nodes does not try to find the whole path from source to sink, instead they only assess, which neighbor is best as their next hop.
- Whenever, a path is not best anymore, it can be changed by just changing next hop of one or more node in the path. The time taken in this process is very less as it includes only few broadcasts.
- There are no network wide broadcasts, hence it may result in less number of control plackets in the network.

IV. Results and Discussion

The simulation of proposed routing protocol is implemented in ns-2. The results are compared with Adhoc Ondemand Distance Vector routing (AODV) protocol.

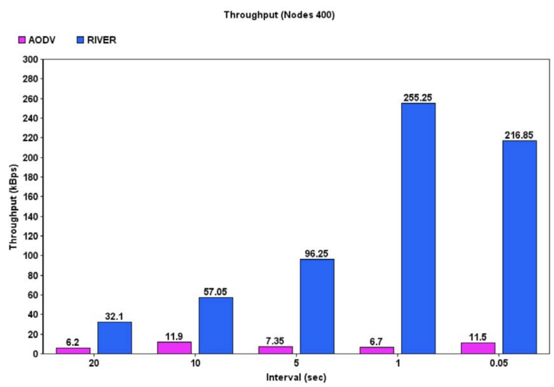

Fig. 6. Comparison of River and AODV on throughput for 400 nodes.

A. Comparison of River and AODV for Fixed Topology

For comparison of River and AODV, we have used a fixed topology of 400 nodes and varied packet transmission interval. The interval chosen are 20,10,5,1 and 0.05. There is one sink node and remaining 399 are sensor nodes. The sink node has been placed at an approximate centre of the network.

1) Comparison of River and AODV on Throughput for 400 nodes: Figure 6 shows the comparison of River and AODV for throughput. Throughput refers to the number of data bytes received in a second at the sink node. In this graph the pink bar represents AODV and the blue bar represents River. For interval 20, throughput of AODV is 6.2 and of River is 32.1. River throughput is 80.68% better than that of AODV. For interval 10, throughput of AODV is 11.9 and of River is 57.05. River throughput is 79.14% better than that of AODV. For interval 5, throughput of AODV is 7.35 and of River is 96.25. River throughput is 92.36% better than that of AODV. For interval 1, throughput of AODV is 6.7 and of River is 255.25. River throughput is 97.37% better than that of AODV. For interval 0.05, throughput of AODV is 11.5 and of River is 216.85. River throughput is 94.69% better than that of AODV. We also observed that, the best performance of AODV is at interval 10 with a throughput value of 11.9 KBps. Whereas, the best performance of River is at interval 1 with 255.25 KBps throughput. From above observations, it is clear that, the performance of River in terms of throughput is better than that of AODV.

2) Comparison of River and AODV Normalized Routing Load (NRL)for 400 nodes: Figure 7 shows the comparison of River and AODV for NRL. NRL refers to number of routing packets(received at any hop) per data packet (reached at destination). In this graph the maroon bar represents AODV an blue bar represents River. For interval 20, NRL of AODV is 3520.386 and of River is 8.537. AODV NRL is 813 times than that of River or River NRL is only 0.24% of AODV NRL. For interval 10, NRL of AODV is 1935.787 and of River is 4.806. AODV NRL is 403 times than that of River or River NRL is only 0.25%of AODV NRL. For interval 5, NRL of AODV is 4430.235 and of River is 2.91. AODV NRL is 1522 times than that of River or River NRL is only 0.07% of AODV NRL. For interval 1, NRL of AODV is 3410.789 and of River is 1.174. AODV NRL is 2905 times than that of River or River NRL is only 0.03% of AODV NRL. For interval 0.05, NRL of AODV is 332.568 and of River is 1.68. AODV NRL is 198 times than that of River or River NRL is only 0.5% of AODV NRL. We have also observed that, the worst performance of AODV is at interval 5 with an NRL value of 4430.235. This means that to deliver a data packet from source to the destination, a total of 4430.235 routing packets are required. Whereas, the worst performance of River is at interval 20 with an NRL value of 8.537. It means, to send a data packet from source to destination , only 8.537 routing packets were required. From above observations, it is clear that, the performance of River in terms of NRL is also better than that of AODV.

3) Comparison of River and AODV on Packet Delivery Fraction (PDF) for 400 nodes: Figure 8 shows the comparison of River and AODV for PDF. PDF is the ratio of the packet received at the sink node to the total number of packets sent in the network. In this graph, the green bar represents AODV and blue bar represents River. The x-axis represents the packet transmission interval and the y-axis represents the PDF. By multiplying the PDF value by 100 we can get results in percentage. For interval 20, PDF of AODV is 0.1833 and of River is 0.9028. River PDf is 393% better than that of AODV. For interval 10, PDF of AODV is 0.1868 and of River is 0.8507. River PDF is 78.45% better than that of AODV. For interval 5, PDF of AODV is 0.0678 and of River is 0.7413. River PDF is 90.85% better than that of AODV. For interval 1, NRL of AODV is 0.0228 and of River is 0.4088. River PDF is 1693% better than that of AODV. For interval 0.05, PDF of AODV is 0.0133 and of River is 0.0293. River PDF is 54.60% better than that of AODV. We have also observed that, the best performance of AODV is at interval 10 with an PDF value of 0.1868. This means that, of all the data packet sent by the sensors only 19% (round off(0.1868 * 100)) packet reached at sink node. Whereas, the best performance of River is at interval 20 with an PDF value of 9028. It means, of all the data packet sent by the sensors, as good as 90% packet reached at sink node. From above observations, it is clear that, the performance of River in terms of PDF also, is better than that of AODV.

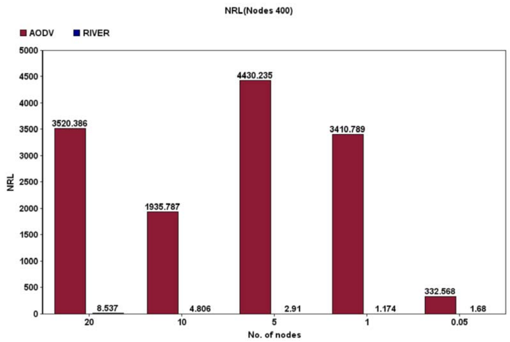

Fig. 7. Comparison of River and AODV normalized routing load for 400 nodes.

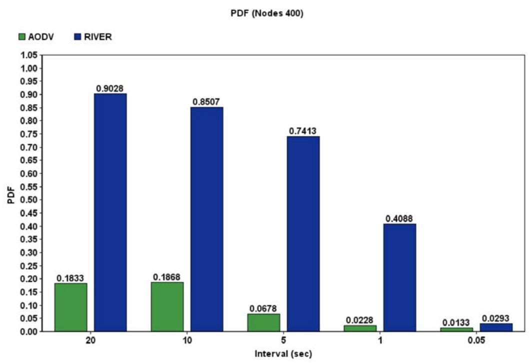

Fig. 8. Comparison of River and AODV on packet delivery fraction for 400 nodes.

B. Comparison of River and AODV for Varying Topology

In this section, we will present the comparison result of AODV and River for fixed packet transmission interval of 0.05 seconds and with varying topology of nodes 100,200,300 and 400. The packet transmission interval refers to the time between two successive packet transmissions from source sensor node.The packet transmission interval of 0.05 seconds, signifies that every second, the source sensor nodes are transmitting 20 (1/interval) data packets. For high data rate WSN, this interval is less, whereas, for low data rate WSN, this interval is more. Hence, we are showcasing our results for a lesser interval of 0.05 rather that an interval of 10 or 20. These results shows the behaviour of both the protocol under HWSN conditions.

1) Comparison of River and AODV on Throughput for 0.05 interval: Figure 9 shows the comparison of River and AODV for throughput. In this graph, the yellow bar represents AODV and blue bar represents River. For 100 nodes, throughput of AODV is 41.85 and of River is 121.85. River throughput is 65.65% better than that of AODV. For 200 nodes, throughput of AODV is 24.85 and of River is 88.2. River throughput is 71.82% better than that of AODV. For 300 nodes, throughput of AODV is 23.55 and of River is 76.7. River throughput is 69.29% better than that of AODV. For 400 nodes, throughput

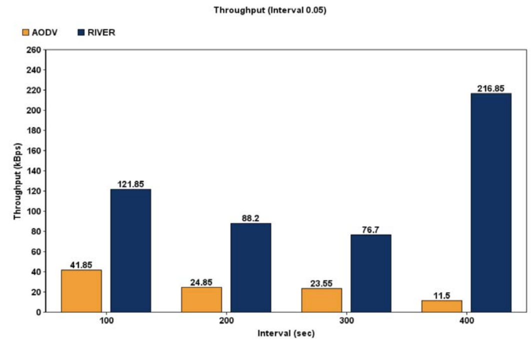

Fig. 9. Comparison of River and AODV on throughput for 0.05 interval

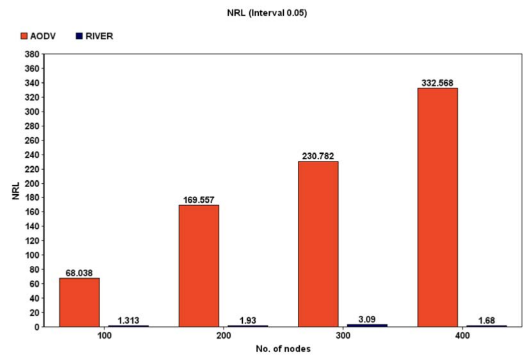

Fig. 10. Comparison of River and AODV normalized routing load for 0.05 interval

of AODV is 11.5 and of River is 216.85. River throughput is 94.69% better than that of AODV. We observed that, the best performance of AODV is 121.85 with 100 nodes. Whereas, the best performance of River is 216.85 with 400 nodes. This graph shows us that, the performance of AODV is constantly decreasing with an increase in the number of nodes. But the performance of River decreased initially, but again increased at nodes 400. It shows that the River protocol is more scalable that AODV. The result shows performance of River with in terms of PDF is better than AODV in varying topology also.

2) Comparison of River and AODV Normalized Routing Load for 0.05 interval: Figure 10 shows the comparison of River and AODV on NRL. Normalized Routing Load is the ratio of routing packets to the data packets received. The orange bar is representing AODV and the blue bar representing River protocol. For 100 nodes, NRL of AODV is 68.038 and of River is 1.133. AODV NRL is 60 times than that of River or River NRL is only 2%of the AODV NRL. For 200 nodes, NRL of AODV is 169.557 and of River is 1.93. AODV NRL is 88 times than that of River or River NRL is only 1%of the AODV NRL. For 300 nodes, NRL of AODV is 230.782 and of River is 3.09. AODV NRL is 75 times than that of River or River NRL is only 1% of the AODV NRL. For 400 nodes, NRL of AODV is 332.568 and of River is 1.68. AODV NRL is 198 times than that of River or River NRL is only 0.5% of the AODV. The of AODV performed worst with 400 nodes giving

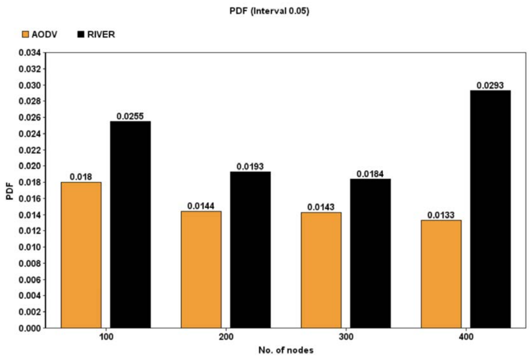

Fig. 11. Comparison of River and AODV on packet delivery fraction for 0.05 interval.

NRL of 332.567 , whereas, the worst performance of River is with 300 nodes, thats too, only giving an NRL of 3.09. The River NRL is absolutely negligible in front of AODV's NRL. From above observations, it is clear that, the performance of River in terms of NRL also, is better than that of AODV.

3) Comparison of River and AODV on Packet Delivery Fraction for 0.05 interval: Figure 11 shows the comparison of River and AODV for packet delivery fraction. PDF is the ratio of the packet received at the sink node to the total number of packets sent in the network. The yellow bar in the graph represents AODV and the black bar represents River. For 100 nodes, PDF of AODV is 0.0183 and of River is 0.0255. River PDF is 39% better than that of AODV. For 200 nodes, PDF of AODV is 0.0144 and of River is 0.0193. River PDf is 34% better than that of AODV. There is a 21%decrease in the PDF when we increased the number of nodes from 100 to 200. but it increases to 24% in case of River. For 300 nodes, PDF of AODV is 0.0143 and of River is0.0184. River PDF is 29% better than that of AODV. there is a 0.7% decrease in the PDF when we increased the number of nodes from 100 to 200. But there is 4% decrease for River. For 400 nodes, PDF of AODV is 0.0133 and of River is 0.0293. River PDF is 120%better than that of AODV. There is a 7% decrease in the PDF, when we increased the number of nodes from 300 to 400. Whereas, there is a 60% increase in case of River. We can also observe that, the best performance 0.018 of AODV. The River protocol shows its nest performance of 0.0293 with 400 nodes, where the performance of AODV is worst(0.0133). From above observations, it is clear that, the performance of River in terms of PDF also, is very much better that that of AODV.

V. Conclusions

We have Designed a new routing protocol, called River; which is also a bio-inspired algorithm. This algorithm is based on the Intelligent Water Drops (IWD) algorithm.The simulation results shows the better performance of River over AODV. The reason for better performance of River over AODV is that, the Normalized Routing Load has been significantly reduced in it. The future work for this protocol may include the functionalities to tackle some problems which might occur during real time implementation of the protocol, eg. link failure control mechanism, packet drop control mechanism, dead node (node which has consumed all of its energy) detection and tackle mechanism etc. We can look into the appropriate inclusion of error and acknowledgement packets to make the service more robust and reliable. A balanced approach is required to decide for which situations we want to send the acknowledgement packets, because if we are not very careful in this decision, it will substantially increase the routing load of the protocol.

References

[1] Y. Al-Obaisat and R. Braun, "On wireless sensor networks: architectures, protocols, applications, and management," 2007.
[2] M. Saleem, G. A. Di Caro, and M. Farooq, "Swarm intelligence based routing protocol for wireless sensor networks: Survey and future directions," Information Sciences, vol. 181, no. 20, pp. 4597–4624, 2011.
[3] J. Nikodem, M. Nikodem, M. Woda, R. Klempous, and Z. Chaczko, Relation-based message routing in wireless sensor networks. InTech, 2010.
[4] S. Binitha and S. S. Sathya, "A survey of bio inspired optimization algorithms," International Journal of Soft Computing and Engineering, vol. 2, no. 2, pp. 137–151, 2012.
[5] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," Advanced engineering informatics, vol. 19, no. 1, pp. 43–53, 2005.
[6] E. Bonabeau, M. Dorigo, and G. Theraulaz, Swarm intelligence: from natural to artificial systems. Oxford university press, 1999, no. 1.
[7] H. Shah-Hosseini, "Intelligent water drops algorithm: A new optimization method for solving the multiple knapsack problem," International Journal of Intelligent Computing and Cybernetics, vol. 1, no. 2, pp. 193–212, 2008.
[8] M. Dorigo and M. Birattari, "Ant colony optimization," in Encyclopedia of machine learning. Springer, 2010, pp. 36–39.
[9] G. Di Caro and M. Dorigo, "Antnet: Distributed stigmergetic control for communications net- works," Journal of Artificial Intelligence Research, pp. 317–365, 1998.
[10] M. Othman, S. Senthilkumar, and X. Yi, "Fourth international conference on digital image processing (icdip 2012)," in Proc. of SPIE Vol, vol. 8334, 2012, pp. 833 401–1.
[11] T. Camilo, C. Carreto, J. S. Silva, and F. Boavida, "An energy-efficient ant-based routing algorithm for wireless sensor networks," in Ant Colony Optimization and Swarm Intelligence. Springer, 2006, pp. 49–59.
[12] W. Cai, X. Jin, Y. Zhang, K. Chen, and R. Wang, "Aco based qos routing algorithm for wireless sensor networks," in Ubiquitous Intelligence and Computing. Springer, 2006, pp. 419–428.
[13] Y. Sun, H. Ma, L. Liu, and Y. Zheng, "Asar: An ant-based service-aware routing algorithm for multimedia sensor networks," Frontiers of Electrical and Electronic Engineering in China, vol. 3, no. 1, pp. 25–33, 2008.
[14] A. Thakare and L. Malik, "Design approach of self-organized routing protocol in wireless sensor networks using biologically inspired methods," in Proceedings of Ninth International Conference on Wireless Communication and Sensor Networks. Springer, 2014, pp. 203–220.
[15] B. Krishnamachari, D. Estrin, and S. Wicker, "Modelling data-centric routing in wireless sensor networks," in IEEE infocom, vol. 2, 2002, pp. 39–44.
[16] R. GhasemAghaei, A. Rahman, M. A. Rahman, and W. Gueaieb, "Ant colony-based many-to- one sensory data routing in wireless sensor networks," in Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on. IEEE, 2008, pp. 1005–1010.