Research Journal of Applied Sciences, Engineering and Technology 11(11): 1179-1189, 2015
DOI: 10.19026/rjaset.11.2224
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: April 7, 2015 Accepted: July 14, 2015 Published: December 15, 2015

Research Article

Fuzzy Based Congestion Control for Backpressure Routing Algorithm in Wireless Sensor Networks

[1]C.J. Raman and [2]Visumathi James
[1]Department of Computer Science and Engineering, Sathyabama University,
Department of Information Technology, St. Joseph's College of Engineering,
[2]Department of Computer Science and Engineering,
Jeppiaar Engineering College, OMR, Chennai-119, Tamilnadu, India

Abstract: The main aim of this study is to detect congestion and provide efficient route recovery mechanism using back pressure technique. In this study, we propose a fuzzy based congestion control and backpressure routing technique in wireless sensor networks. In the Fuzzy based congestion control technique, Fuzzy logic decision model is used to estimate the congestion status of each node based on the parameters number of contenders, buffer occupancy percentage of parent nodes and traffic load. In cluster based backpressure routing, clusters are formed and cluster heads are elected based on the congestion status of the nodes. By simulation results, we show that the proposed algorithm reduces the overhead and increases the routing efficiency.

Keywords: Back pressure routing algorithm, congestion control, fuzzy logic, wireless sensor network

INTRODUCTION

Wireless Sensor Network (WSN): A wireless sensor network comprises autonomous, tiny and cheap wireless sensor nodes in a physical phenomenon and deployed remotely like high mountain area and satellite in the outer space. The embedded sensors continuously monitor the physical or environmental process like temperature, sound, vibration, pressure, motion, or pollutants etc and send this information in a multi-hop fashion to a special node called the sink (Fig. 1). WSN characteristics are centralized data collection, multi-hop data transmission and many-to one traffic patterns. WSN framework has a various applications of deploying sensor networks like environmental monitoring, battlefield surveillance, health and industrial monitoring control, habitat monitoring, structural health monitoring, image sensing and physical game etc (Gadze *et al*., 2013; Yang *et al*., 2013).

The nodes closer to the base stations require to send more data packets resulting in severe traffic burden. This causes severe packet collisions, network congestion and packet loss. In most severe cases it even results in congestion collapse. The physical parameters and the information of the interactions, along with variable wireless network conditions, may lead to unpredictable traffic load variations and the link capacity fluctuations (Gadze *et al*., 2013; Yang *et al*., 2013).

Congestion control in WSN: Congestion, a major issue in wireless sensor networks results in packet losses and increased transmission latency which affects energy efficiency and application QoS and hence must be efficiently controlled. Congestion also causes indiscriminate data dropping resulting in increased energy consumption to route packets that will be dropped downstream as links become saturated. Congestion occurrence in sensor networks is of two types: Node level congestion and Link level congestion. The former occurs due to buffer overflow in the node which leads to packet loss and increased queuing delay. Packet loss results in retransmission and hence consumes additional energy. Link-level congestion is related to the wireless channels shared by several nodes using protocols, such as CSMA/CD (carrier sense multiple access with collision detection). When multiple active sensor nodes try to seize the channel simultaneously, collisions could occur. Link level congestion increases packet service time and decreases both link utilization and overall throughput and wastes energy at the sensor nodes. Both node level and link level congestions have direct impact on energy efficiency and QoS (Agarwal, 2013).

Congestion control is a hard task for wireless networks since identifying congestion is not as simple as in wired networks. In general, congestion control

Corresponding Author: C.J. Raman, Department of Computer Science and Engineering, Sathyabama University, Department of Information Technology, St. Joseph's College of Engineering, OMR, Chennai-119, Tamilnadu, India

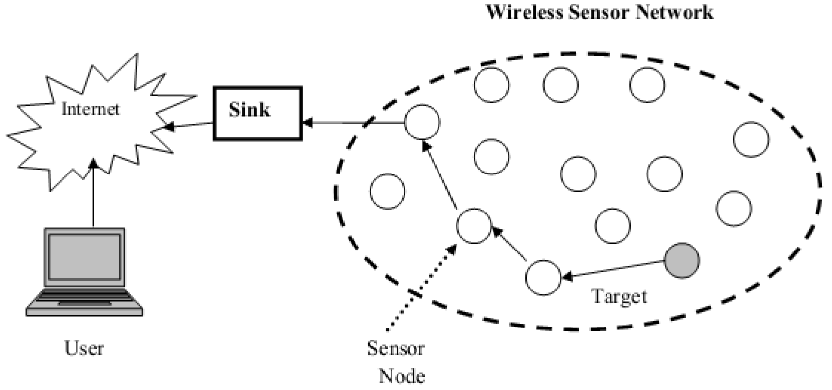

Fig. 1: Structure of wireless sensor network

protocols for WSNs consider specific parameters like local buffer occupancy, packet arrival rate and packet service time. However, determining the level of congestion based on a specific parameter may give incorrect results (Sonmez *et al*., 2014).

Back pressure routing in WSN: The Backpressure Routing algorithm will not compute routes for packets initially. But on reception of requests of data transfer from node, the algorithm verifies the current congestion value and then computes the routes accordingly (Ravi *et al*., 2012).

The back-pressure algorithm, a famous throughput-optimal algorithm whose implementation needs each node to maintain a separate queue for each commodity in the network and only one queue is served at a time. This leads to bad delay performance even though the traffic load is not close to network capacity. In addition, maintaining queuing data structure each node is complex due to the higher number of commodities (Agarwal, 2013). The approach generates queue backlog gradients that decrease towards the sink, where these queue backlogs encode certain utility and penalty information. With information about queue backlogs and link states, nodes can make source rate, packet routing and forwarding decisions without the notion of end-to-end routes (Moeller *et al*., 2011).

Maintaining up-to-date information about neighbor backpressure is critical to ensure good routing performance (Nori *et al*., 2014).

There are several issues in implementing Back pressure mechanisms practically at the routing layer:

- Without careful defining of the link weights, backpressure routing is affected due to either excessively high hop-counts or, at the other extreme, over emphasizes low hop counts, leading to wasted transmissions and link-layer packet losses.
- Backpressure routing can suffer from inordinately large delays due to large queue sizes they should be maintained to provide a gradient for data flow.
- Queues grow in size with distance from the sink which forms a issue in large-scale deployments due to maximum queue size limitations in resource-constrained devices (Moeller *et al*., 2011).

Problem identification and objectives: Nori *et al.* (2014), congestion detection in WSN was provided by using fuzzy logic system by considering number of contenders, buffer occupancy percentage of parent nodes and traffic load as the inputs. The output of the fuzzy logic system denotes the congestion status which is obtained from the fuzzy rules and then defuzzified later. However it does not involve route recovery mechanism and has no support for back pressure technique to inform sensor nodes about congestion.

Hence the main objective of this study is to design efficient route recovery meachnism using back pressure technique such that congestion is mitigated immediately.

LITERATURE REVIEW

Majidi and Mirvaziri (2014) presented a congestion control method by using backpressure routings and service differentiation in WMSNs. This algorithm will not pre calculate routes and next step is selected dynamically. Each node monitors its own queue backlog and its neighbor's queue backlog and varies its own rate and chooses routes according to queue backlog of its neighbors. Dynamic prioritization was deployed for service differentiation if there are two or more data with same condition in backpressure routing. However packet delivery ratio was decreased.

Venkataraman *et al.* (2009) applied a Vector Auto Regression (VAR) based trust model over the Backpressure Collection Protocol (BCP), a collection mechanism based on dynamic backpressure routing in Wireless Sensor Networks (WSN). Then an auto regression based scheme was deployed to embed trust into the link weights, making it more likely for trusted links to be scheduled. However throughput is decreased.

Sonmez *et al.* (2014) proposed a fuzzy logic-based congestion estimation approach to detect congestion efficiently. Three different congestion indicators were joined to get a more accurate measurement of the congestion. A novel congestion mitigation technique was deployed to perform quality adaptation on-the-fly and provided a considerable frame delivery and latency performance gain. A protocol was deployed utilizing the cross-layer functionalities and utilized cross-layer information exchange among the application, transport, MAC and routing layers. However average frame loss ratio was increased.

Dvir and Buttyan (2011) presented and implemented a recovery algorithm according to a weighted backpressure function that bypasses the jamming area by spreading the congestion over a large subset of the sensor nodes, when no tree reconstruction and mapping of the jamming area are needed.

Nori *et al*. (2014) designed a scheme for integrating backpressure queues into the IP stack. A new queue-based adaptive beaconing mechanism was utilized to reduce network bandwidth and CPU load on the nodes. NULL packets, a special control packet, were adapted to integrate with the IP architecture. Queue information was spread across the network in standard-compliant ways. The protocol to efficiently co-ordinate data and control information across software layers within the node was implemented finally.

Moeller *et al*. (2011) presented the Backpressure Collection Protocol (BCP) for sensor networks, been the first implementation of dynamic backpressure routing in wireless networks. The traditional FIFO queue service in backpressure routing was replaced with LIFO queues to reduce the average end-to-end packet delays for delivered packets drastically. In addition, backpressure scalability was improved by introducing a new concept of floating queues into the backpressure framework.

Sridharan *et al*. (2008) evaluated a simple backpressure scheduling policy enable nodes to transmit much distance due to a positive queue differential (irrespective of its size) gives performance comparable to more sophisticated heuristics. The results show sensor networks do not require modifications to the MAC to implement backpressure based protocols. A comparative evaluation of backpressure protocols against protocols optimized for wireless networks was done to explain no single parameter value guaranteeing optimal performance was present to backpressure based rate control protocols for a given topology.

PROBLEM IDENTIFICATION AND SOLUTION

Overview: In this study, we propose a fuzzy based congestion control and backpressure routing technique in wireless sensor networks. Initially, the clusters are formed and cluster heads are elected as per the congestion status of the node. The congestion status of the node is estimated using fuzzy logic technique based on the parameters number of contenders, buffer occupancy percentage of parent nodes and traffic load. When source wants to route data to the destination node, it uses the cluster based back pressure routing

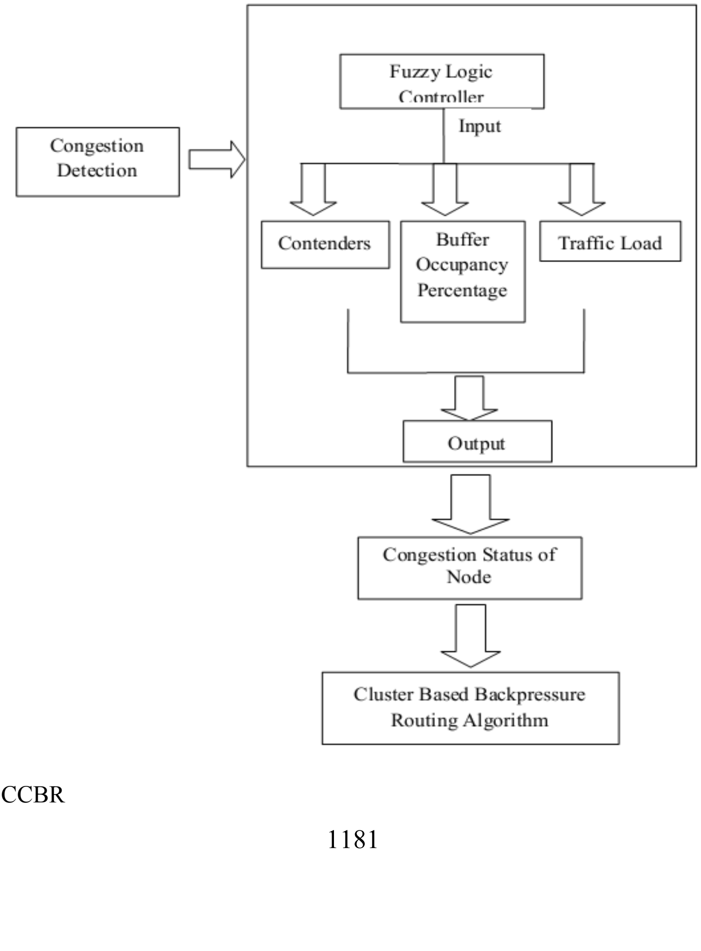
Congestion Detection
Fuzzy Logic Controller
Input
Contenders
Buffer Occupancy Percentage
Traffic Load
Output
Congestion Status of Node
Cluster Based Backpressure Routing Algorithm

Fig. 2: Block diagram of FBCCBR

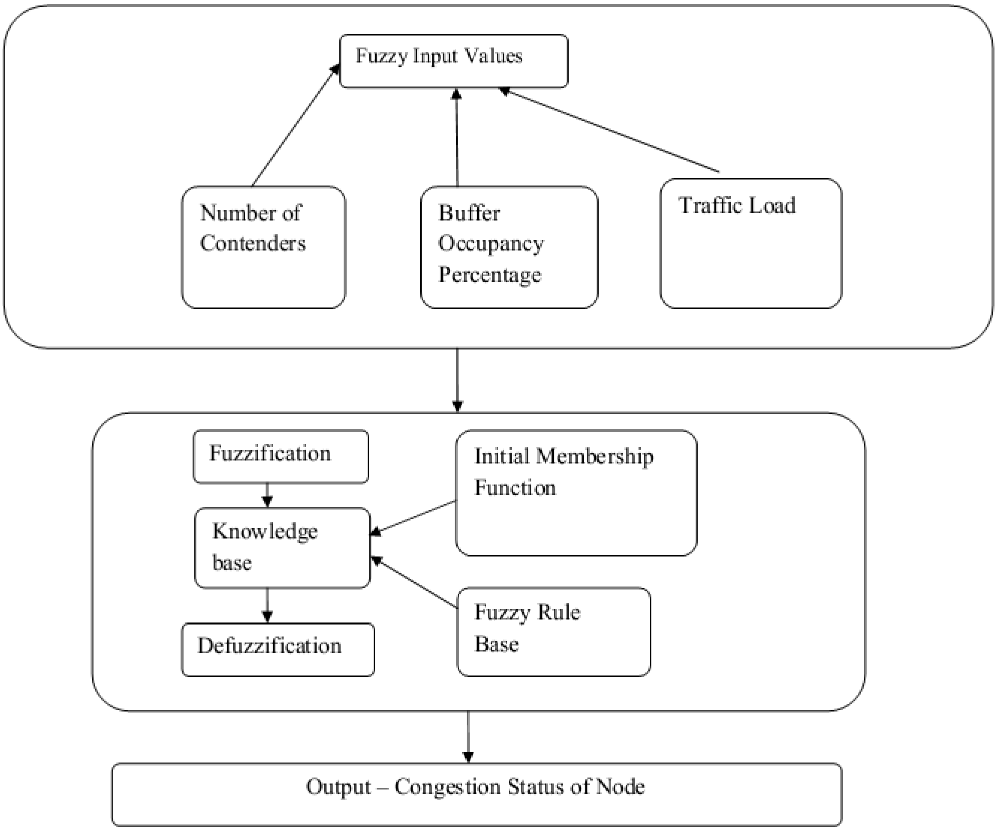

Fig. 3: Fuzzy inference system

technique where the cluster head acts as the gateway node for data transmission. The weighed back pressure function is calculated for each neighbor and chooses the neighbor with high value. The weighed back pressure function consists of the congestion status obtained from the output of Fuzzy logic, difference between nodal levels and its candidate neighbor, the difference between the node and its candidate neighbor's queues and the routing status.

Figure 2 shows the block diagram of the proposed Fuzzy based congestion control and Backpressure routing (FBCCBR) algorithm.

Fuzzy based congestion detection: We detect the congestion status of the node using fuzzy logic technique. The parameters number of contenders, buffer occupancy percentage of parent nodes and traffic load are taken as input for the fuzzy member ship functions. These inputs are fuzzified and processed by interference system to provide congestion status of the node as output which is then de-fuzzified. The outputs are aggregated and form new fuzzy sets.

The steps that determine the fuzzy rule based interference are as follows:

- **Fuzzification:** This involves obtaining the crisp inputs from the selected input variables and estimating the degree to which the inputs belong to each of the suitable fuzzy set.
    - o **Number of contenders:** It is estimated using the RTS or CTS packets which are generated by the neighbor nodes. If there are too many contenders, collision probability is higher.
    - o **Buffer occupancy percentage:** If the buffer occupancy percentage of the node is high, the congestion probability will also be high.
    - o **Traffic load:** It is defined as the ratio of the incoming packets to the outgoing packets
- **Rule evaluation:** The fuzzified inputs are taken and applied to the antecedents of the fuzzy rules. It is then applied to the consequent membership function.
- **Aggregation of the rule outputs:** This involves merging of the output of all rules.
- **Defuzzification:** The merged output of the aggregate output fuzzy set is the input for the defuzzification process and a single crisp number is obtained as output.

The fuzzy inference system is illustrated using Fig. 3.

Fuzzification: This involves fuzzification of input variables such as number of contenders (C), buffer occupancy percentage (B) and traffic load (T) and these

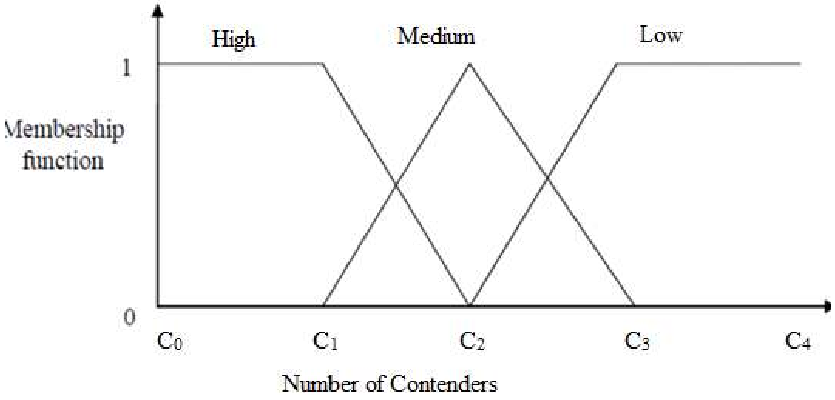

Fig. 4: Membership function of number of contenders

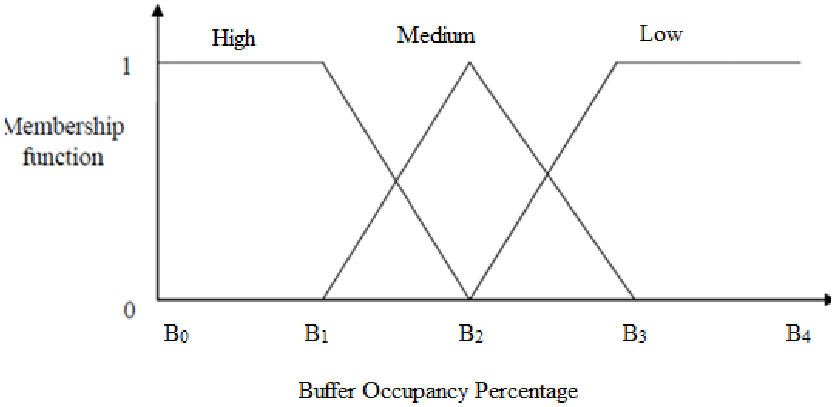

Fig. 5: Membership function of buffer occupancy percentage

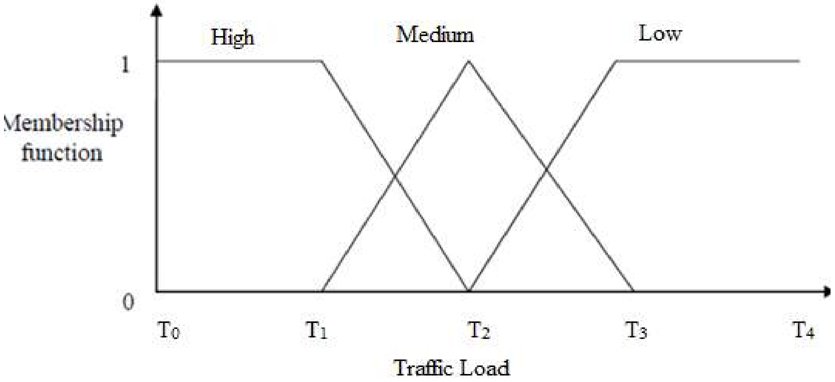

Fig. 6: Membership function of traffic load

inputs are given a degree to appropriate fuzzy sets. The crisp inputs are combination of C, B and T. We take the possibilities, high, medium and low for C, B and T.

Figure 4 to 7 shows the membership function for the input and output variables. Due to the computational efficiency and uncomplicated formulas, the triangulation functions are utilized which are widely utilized in real-time applications. Also a positive impact is offered by this design of membership function.

In table C, B and T are given as inputs and the output represents the Congestion Status (CS). The fuzzy sets are defined with the combinations presented in Table 1.

Table 1 demonstrates the designed fuzzy inference system. This illustrates the function of the inference engine and method by which the outputs of each rule are combined to generate the fuzzy decision.

For example:

Let us consider Rule 26.

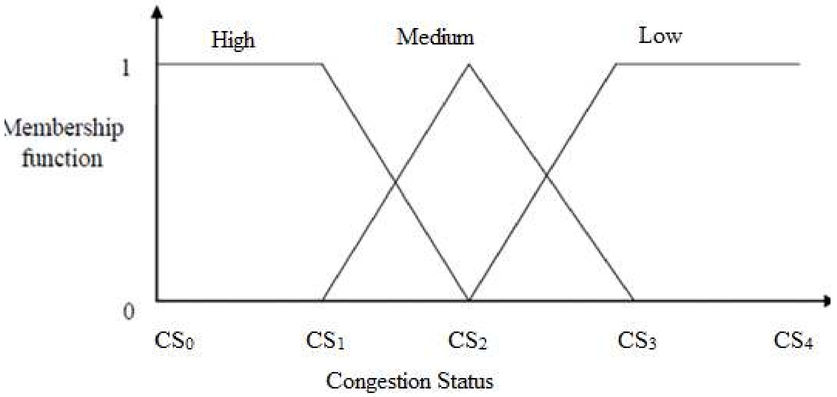

Fig. 7: Membership function of congestion status

Table 1: Fuzzy rule base

S. No	C	B	T	CS
1	Low	Low	Low	Low
2	Low	Low	Medium	Low
3	Low	Low	High	Medium
4	Low	Medium	Low	Low
5	Low	Medium	Medium	Medium
6	Low	Medium	High	Medium
7	Low	High	Low	Low
8	Low	High	Medium	Medium
9	Low	High	High	Medium
10	Medium	Low	Medium	Medium
11	Medium	Low	High	Medium
12	Medium	Medium	Low	Low
13	Medium	Medium	Medium	Medium
14	Medium	Medium	High	Medium
15	Medium	High	Low	Medium
16	Medium	High	Medium	Medium
17	Medium	High	High	Medium
18	Medium	Low	Low	Low
19	High	Low	Medium	Medium
20	High	Low	High	Medium
21	High	Medium	Low	Medium
22	High	Medium	Medium	High
23	High	Medium	High	High
24	High	High	Low	High
25	High	High	Medium	High
26	High	High	High	High

```
If (C, B, T = high)
Then
CS = high
End if
```

This reveals that the selected node is subject to congestion.

Defuzzification: Defuzzification is used for extracting a crisp value from a fuzzy set as a representation value. We consider the centroid of area strategy for defuzzification.

$$F_{QoS} = \frac{\int_f \eta_{agg}(F)_{fdf}}{\eta_{agg}(F)_{df}} \tag{1}$$

where,
$\eta_{agg}(F)$ = Aggregated output of membership function

Cluster based back pressure routing algorithm: When the sensor nodes are deployed in the network, based on the fuzzy logic, the nodes with minimum congestion status is chosen as Cluster Head (CH). CH then acts as the gateway nodes for routing the data and cluster members' acts as interior nodes.

Let C(i) be the cluster that includes N_i
N_i can either be the gateway node or interior node.
N_i is said to be gateway of C (i), if there exists a node N_j such that $N_j \notin C$ (i) and $(N_i, N_j) \in V$. V represents the network link.
N_i is said to be interior node of C (i), if N_i is not a gateway node.
Let X_i be the set of all gateway and interior nodes in the cluster that includes N_i except N_j itself
i.e., $X_i = G_i \, U \, I_i \setminus \{N_i\}$
Let S and D be the source and destination node respectively.
Let $K_S^D[t]$ $(S, D \in N_i)$ be the rate at which a flow [S, D] generates packets at time t.
Let R_{tx} [t] be the transmission rate over link (N_i, N_j) at time t.

Let $x_g^D[t]$ and $\hat{x}_g^D[t]$ be the length of regulated and real queue respectively.

The steps involved in cluster based back-pressure routing algorithm are as follows:

- At time t, S splits the external data flow K_S^D into $\{K_S^{g,D}[t]\}_{g \in \{0\} U G(D)}$ such that

$$\{K_S^{g,D}[t]\} = \begin{Bmatrix} K_S^D, & g = g* \\ 0, & g \neq g* \end{Bmatrix} \tag{2}$$

where, $g* = \arg \min\limits_{g \in \{0\} U G(D)} X_S^{g,D}[t]$

Note: g* = least congested node, where g* > 0 and g* = 0 indicates that best node to route the data packets from S to D without using gateway node.

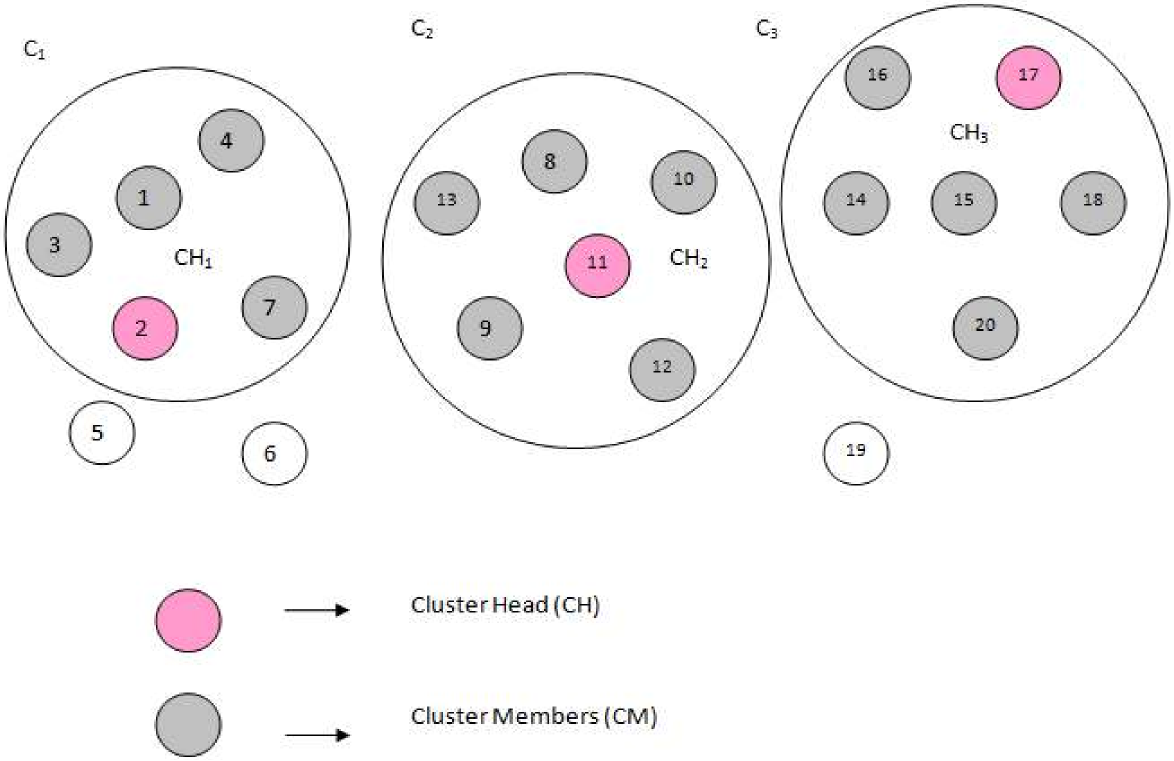

Fig. 8: Cluster formation and CH selection

- After receiving $\{K_S^{g,D}[t]\}_{g\in\{0\}UG(D)}$, S deposits the data packet into queue g at S at time t.
- The gateway g maintains a real queue and regulated queue for each D in the interior of C(g).
- At time t, the rate at which the packets can be transferred from real queue to the regulated queue is estimated using the following Eq. (3):

$$\lambda_g^D[t] = (1+\sigma)\sum_{S:[S.D]\in F} K_S^{g,D}[t] \tag{3}$$

where σ> 0 is a constant which is selected to be small value.

- At time t, the network initially estimates the Back-Pressure Scheduler value $\{W_{ij}\}$ based on the following parameters.
- Congestion status of the node (CS) [estimated in section 3.2]
- Difference between nodal levels and its candidate neighbor (α_{ij})
- Difference between the node and its candidate neighbor's queues (β_{ij})
- Routing status (γ_{ij})

$$W_{ij} \rightarrow F\ \{CS,\ \alpha_{ij},\ \beta_{ij}, \gamma_{ij}\} \tag{4}$$

$$W_{ij} = \max\ \{W_{ij}\} \tag{5}$$

$$W_{ij} = CS_{ij}+\alpha_{ij} + \beta_{ij} + \gamma_{ij} \tag{6}$$

The weighed back pressure function W_{ij} is calculated for each neighbor and chooses the neighbor with high value. Based on the estimated W_{ij} value, N_j transmits the packets in queue to N_i over the link at rate W_{ij}.

The cluster formation and CH selection is illustrated in Fig. 8.

SIMULATION RESULTS

Simulation model and parameters: The Network Simulator (http:///www.isi.edu/nsnam/ns) (NS-2) is used to simulate the proposed architecture. In the simulation, 50 mobile nodes move in a 1000×1000 m region for 20 sec of simulation time. All nodes have the same transmission range of 250 m. The simulated traffic is Constant Bit Rate (CBR).

The simulation settings and parameters are summarized in Table 2.

Performance metrics: The proposed Fuzzy Based Congestion Control and Backpressure Routing (FBCCBR) is compared with the FBCC technique (Sonmez *et al*., 2014). The performance is evaluated mainly, according to the following metrics:

Table 2: Simulation settings and parameters

No. of Nodes	50 and 100
Area size	1000×1000
Mac	IEEE 802.11
Transmission range	250 m
Simulation time	50 sec
Traffic source	CBR
Packet size	512
Rate	50, 75, 100, 125 and 150 kb
Number of flows	1 to 5
Initial energy	8.5J
Transmission power	0.660
Receiving power	0.395

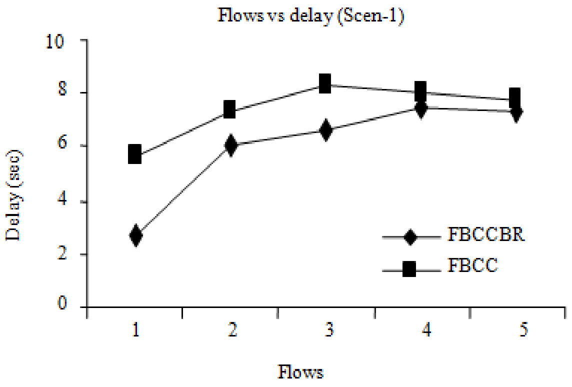

Fig. 9: Flows vs delay

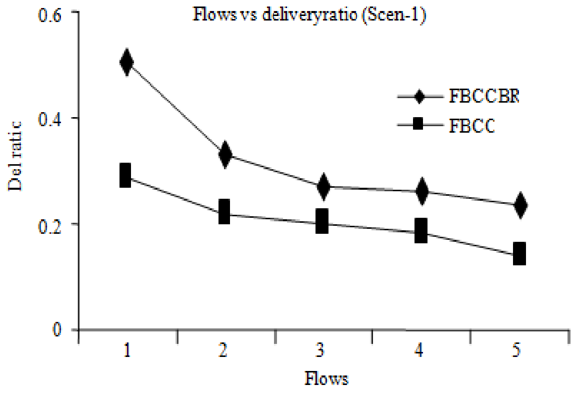

Fig. 10: Flows vs delivery ratio

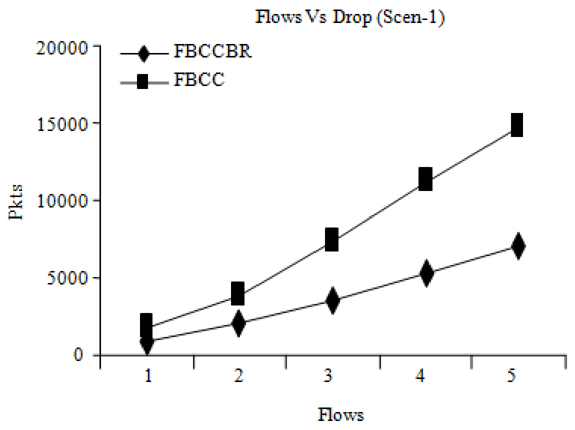

Fig. 11: Flows vs drop

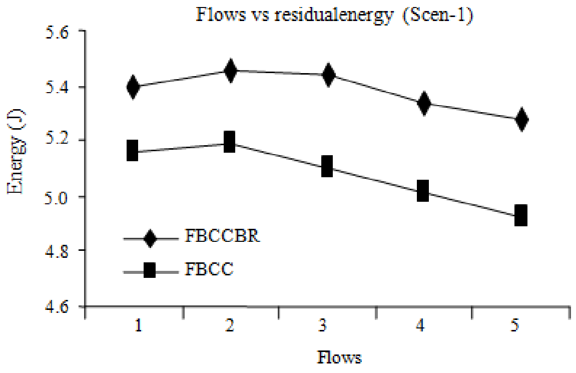

Fig. 12: Flows vs residual energy

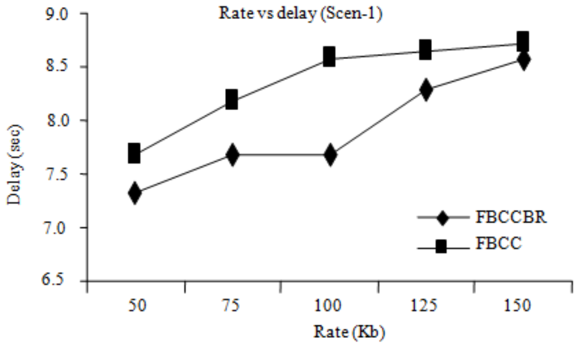

Fig. 13: Rate vs delay

- **Packet delivery ratio:** It is the ratio between the number of packets received and the number of packets sent.
- **Packet drop:** It refers the average number of packets dropped during the transmission
- **Residual energy:** It is the amount of energy remains in the nodes.
- **Delay:** It is the amount of time taken by the nodes to transmit the data packets.

Results:
Scenario-1: 50 Nodes: In scenrario-1, the number of nodes is considered as 50 and the above performance metrics are evaluated by varying the number of data flows and data sending rate.

Based on flows: For 50 nodes, the number of flows is varied from 1 to 5 keeping the rate as 50Kb.

Figure 9 to 12 show the results of FBCCBR and FBCC in terms of delay, delivery ratio, packet drop and residual energy by varying the flows from 1 to 5 for 50 nodes. When comparing the performance of the two works, we infer that FBCCBR outperforms FBCC by 20% in terms of delay, 35% in terms of delivery ratio, 52% in terms of drop and 6% in terms of residual energy.

Based on rate: For 50 nodes, the rate is varied from 50Kb to 150Kb keeping the number of flows as 5.

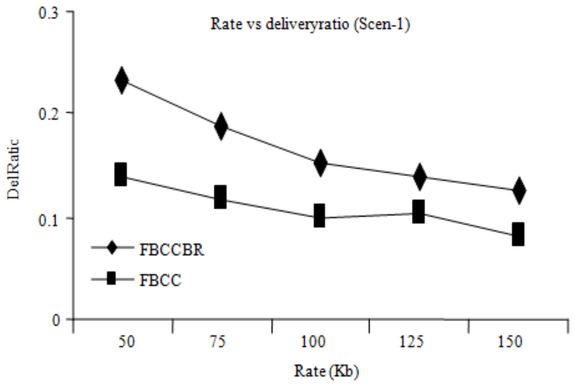

Fig. 14: Rate vs delivery ratio

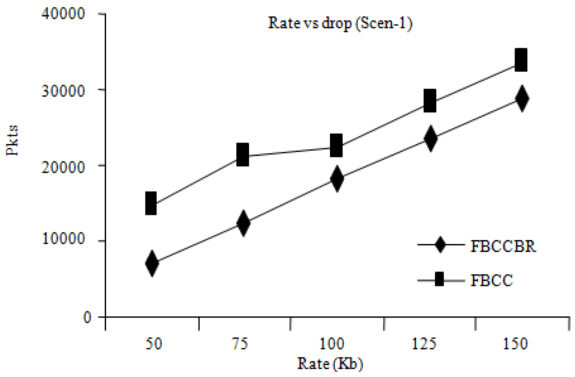

Fig. 15: Rate vs packet drop

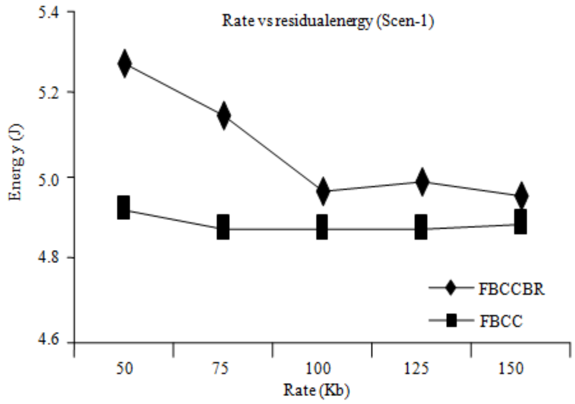

Fig. 16: Rate vs residual energy

Figure 13 to 16 show the results of FBCCBR and FBCC in terms of delay, delivery ratio, packet drop and residual energy by varying the rate from 50Kb to 150Kb for 50 nodes. When comparing the performance of the two works, we infer that FBCCBR outperforms FBCC by 5% in terms of delay, 35% in terms of delivery ratio, 28% in terms of drop and 4% in terms of residual energy.

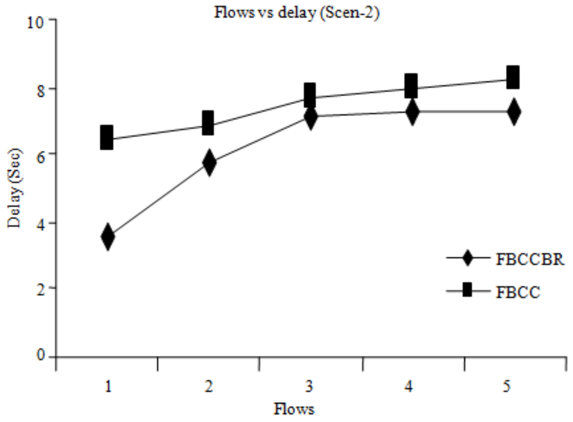

Fig. 17: Flows vs delay

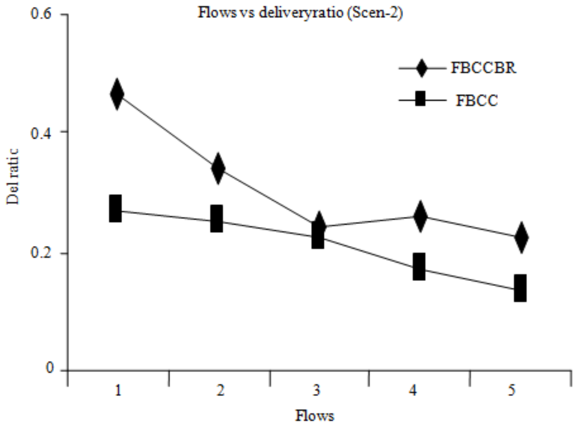

Fig. 18: Flows vs delivery ratio

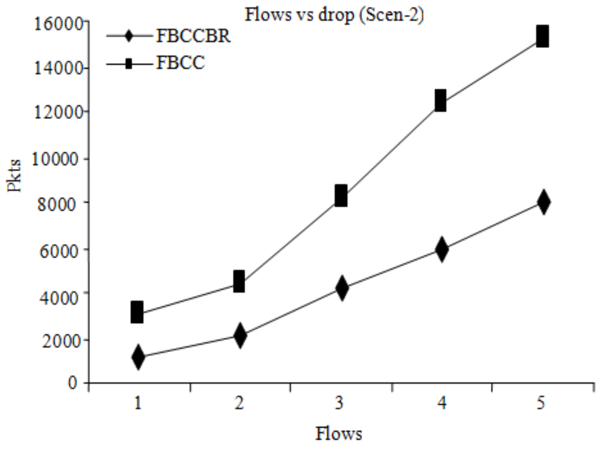

Fig. 19: Flows vs drop

Scenario-2: 100 Nodes: In scenrario-2, the number of nodes is considered as 100 and the performance metrics are evaluated by varying the number of data flows and data sending rate.

Based on flows: For 100 nodes, the number of flows is varied from 1 to 5 keeping the rate as 50Kb.

Figure 17 to 20 show the results of FBCCBR and FBCC in terms of delay, delivery ratio, packet drop and

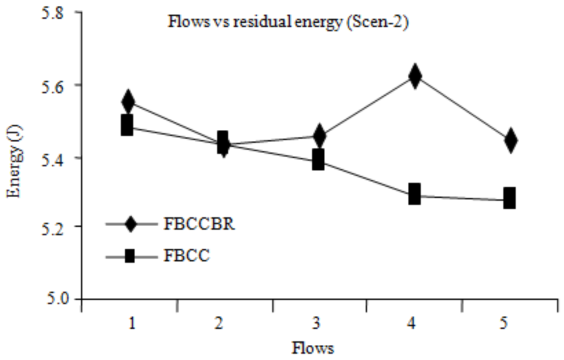

Fig. 20: Flows vs residual energy

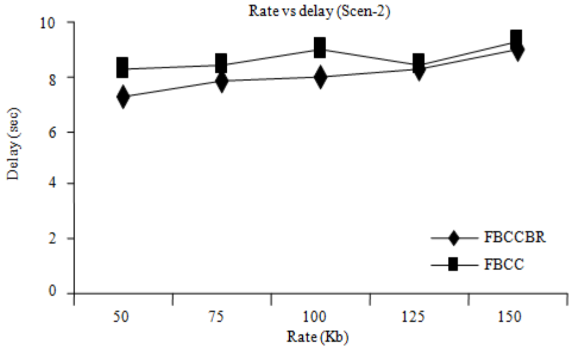

Fig. 21: Rate vs delay

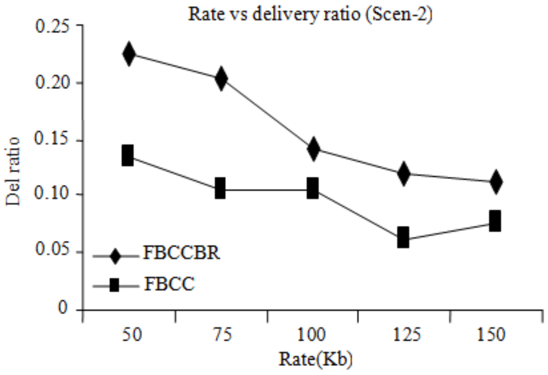

Fig. 22: Rate vs delivery ratio

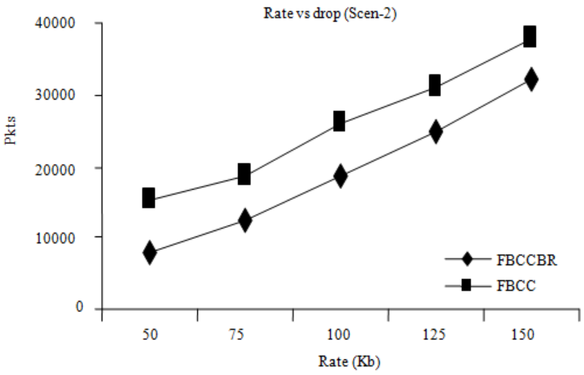

Fig. 23: Rate vs drop

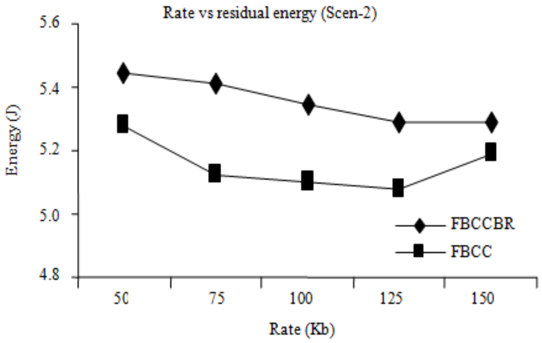

Fig. 24: Rate vs residual energy

residual energy by varying the flows from 1 to 5 for 100 nodes. When comparing the performance of the two works, we infer that FBCCBR outperforms FBCC by 18% in terms of delay, 30% in terms of delivery ratio, 54% in terms of drop and 2% in terms of residual energy.

Based on rate: For 100 nodes, the rate is varied from 50Kb to 150Kb keeping the number of flows as 5.

Figure 21 to 24 show the results of FBCCBR and FBCC in terms of delay, delivery ratio, packet drop and residual energy by varying the rate from 50Kb to 150Kb for 100 nodes. When comparing the performance of the two works, we infer that FBCCBR outperforms FBCC by 7% in terms of delay, 39% in terms of delivery ratio, 29% in terms of drop and 4% in terms of residual energy.

CONCLUSION

In this study, we have proposed a fuzzy based congestion control and backpressure routing technique in wireless sensor networks. Initially, the clusters are formed and cluster heads are elected as per the congestion status of the node. The congestion status of the node is estimated using fuzzy logic technique based on the parameters number of contenders, buffer occupancy percentage of parent nodes and traffic load. When source wants to route data to the destination node, it uses the cluster based back pressure routing technique where the cluster head acts as the gateway node for data transmission. By simulation results, we show that the proposed technique reduces the overhead and increases the routing efficiency.

REFERENCES

Agarwal, P., 2013. Backpressure congestion control technique in wireless sensor network to remove congestion in intelligent transportation system. SRM University, India.

Dvir, A. and L. Buttyan, 2011. Backpressure approach for bypassing jamming attacks in wireless sensor networks. Proceeding of the IEEE Conference on Computer Communications (INFOCOM), Poster Session.

Gadze, J.D., D. Dake and K. Diawuo, 2013. Adaptive Congestion Control Protocol (ACCP) for wireless sensor networks. Int. J. Wirel. Mob. Netw. (IJWMN), 5(5).

Majidi, A. and H. Mirvaziri, 2014. BDCC: Backpressure routing and dynamic prioritization for congestion control in WMSNs. Int. J. Comput. Network Inform. Secur., 5: 29-34.

Moeller, S., A. Sridharan, B. Krishnamachari and O. Gnawali, 2011. Routing without routes: The backpressure collection protocol. Proceeding of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN, 2011), pp: 279-290.

Nori, S., S. Deora and B. Krishnamachari, 2014. BackIP: Backpressure routing in IPv6-based wireless sensor networks. USC CENG Technical Report CENG-2014-01.

Ravi, G., M. Surputheen and R. Srinivasan, 2012. Secure packet transmission in wireless sensor networks using dynamic routing techniques. Int. J. Comput. Appl., 49(22): 0975-8887.

Sonmez, C., O.D. Incel, S. Isik, M.Y. Donmez and C. Ersoy, 2014. Fuzzy-based congestion control for wireless multimedia sensor networks. EURASIP J. Wirel. Comm., 2014: 63, DOI: 10.1186/1687-1499-2014-63.

Sridharan, A., S. Moeller and B. Krishnamachari, 2008. Investigating backpressure based rate control protocols for wireless sensor networks. USC EE CENG Technical Report CENG-2008-7.

Venkataraman, R., S. Moeller, B. Krishnamachari and T.R. Rao, 2009. Trust-based backpressure routing in wireless sensor networks. Int. J. Sensor Network., 17(1): 27-39.

Yang, X., J. Jia, S. Zhang and Z. Li, 2013. Congestion control based on consensus in the wireless sensor network. Int. J. Distrib. Sens. N., 2013: 7, Article ID 632398.