A Radial Basis Neural Network Controller to Solve Congestion in Wireless Sensor Networks

Maab Alaa Hussain

University of Misan/College of Engineering

maab_alaa@yahoo.com

Abstract

In multihop networks, such as the Internet and the Mobile Ad-hoc Networks, routing is one of the most important issues that has an important effect on the network's performance. This work explores the possibility of using the shortest path routing in wireless sensor network . An ideal routing algorithm should combat to find an perfect path for data that transmitted within an exact time. First an overview of shortest path algorithm is given. Then a congestion estimation algorithm based on multilayer perceptron neural networks (MLP-NNs) with sigmoid activation function, (Radial Basis Neural Network Congestion Controller (RBNNCC))as a controller at the memory space of the base station node. The trained network model was used to estimate traffic congestion along the selected route. A comparison study between the network with and without controller in terms of: traffic received to the base station, execution time, data lost, and memory utilization . The result clearly shows the effectiveness of Radial Basis Neural Network Congestion Controller (RBNNCC) in traffic congestion prediction and control.

Keywords: Radial Basis, Congestion, Controller, and memory utilization.

1. Introduction:

These days, the problem of congestion has considered an important matter. the Network is congested when the total instances for resources overtake the total ON-state resources. In other hand, can be said that the network is become in a congestion state when the load increase on the network. This means that the quality of the service (QOS) transferred to the employer decreases. If there is no congestion, when downloading a 1 GB file, the file take a few minutes to downloaded, but the same file gets downloaded in many hours when there is congestion in a network[1]. The Sensor Network can be seen in our lives in different aspects. It is widely used as examples; fire detectors, security sensors, military, health, weather etc. Wireless sensor networks (WSN) have several small devices called sensors, these devices posses the ability to sensing a special phenomenon in the environment, then send the sensed information back to one or several base stations. Elasticity is the primary feature of WSN that makes it unparalleled in terms of mobility of the sensors and the network format. WSN can be created in an areas without any use of wires. Also because the property of wireless sensors which is the freedom to movement along with the self-format, it is an ideal tool for the situations where the sensors are mobile. Because of the above futures, WSN is used in medical applications, military purposes, area monitoring, health application. Challenges in wireless sensor networks come in series with the Elasticity of wireless sensor networks. The sensors should be capable to operate in a manner that can be generate an optimum routing path automatically, and deliver the sensed information back to the base-station. The sensors are dependent on their battery completely to operate, because they are not physically connected to any central unit, also the positions of wireless sensors are not determined prior to the network deployment. Base-station take the received data and treated it then sends the results to the end user or for other treating [2], [3].

Routing protocols at least proactive and reactive categorizes a proactive routing protocol optimized for mobile ad hoc networks, which can also be used for WSN. This protocol facilitates efficient flooding of control messages throughout the network by using selected nodes called Multipoint Relays (MRPs). Reactive routing protocol is intended for Mobile Ad hoc network (MANET) and sensor networks. AODV is a reactive routing protocol. It uses an on-demand approach for finding routes, that is, a route is established only when it is required by a source node for transmitting data packets. AODV has two basic operations: route discovery and route maintenance.[3]

Review of Literature:

- In (2015), Mirza Waseem Hussain Sanjay Jamwal, and Majid Zaman[1], produced a survey of a several congestion techniques about congestion control technique in a computer networks, One of the latest approaches to control the congestion is based on Neural Networks. Which will be useful for new researchers exploring its use in their research problems.
- In (2013), Rajender kumar, et. All [2], they produced that with the growth of wireless communication technology sooner or later it would not be practical or simply physically possible to have a fixed architecture for this kind of network. To obtain good quality in object tracking they used BFO algorithm (even if a sensor node fails, other sensor node can take the responsibility and carry out the tracking Process).

2. Congestion in WSNs:-

Event-based and data stream- based are two groups for WSNs, in terms of collecting and sending data. In both groups, data stream is created, sensed and send from the source devices to the base station. This can lead to the occurrence of congestion in the network. Actually, volatility of sent and received data rate leads to occurrence of congestion in the network. This lead to decreasing throughput of the network and the reliability of sending packets. If the congestion occurs then the lost packets should resent, this may lead to loss of energy and decreasing the network lifetime. From the above, one can notice that the congestion has direct effect on energy efficiency and quality of service. Furthermore, congestion in WSNs reacts with the limitation of node energy, low memory space capacity of nodes, sensitivity to delay and changing the topology of sensors. In order to increase the longevity and reliability of the network, one should control on the congestion [4]. If the capacity of the network is overload, then the congestion will be occurred. Frequent data follow and limited bandwidth could be lead to congestion occurrence. Congestion could be identified using Data arrival rate, channel utilization and queue length. For each node packet loss ratio (PLR) is calculated and the average of all the nodes PLR is estimated and taken as threshold. Those nodes which have PLR less than the threshold value is considered as good node and which are above the threshold value is considered as a bad nodes [5].

3. Congestion control schemes:

Congestion control schemes can be divided into two main categories as follows[6], [7]:
(a) Centralized Congestion Control Schemes: Includes routing protocols assisted with congestion control.
(b) Distributed Congestion Control Schemes: Includes buffer based and cross layer congestion control schemes. See figure (1).

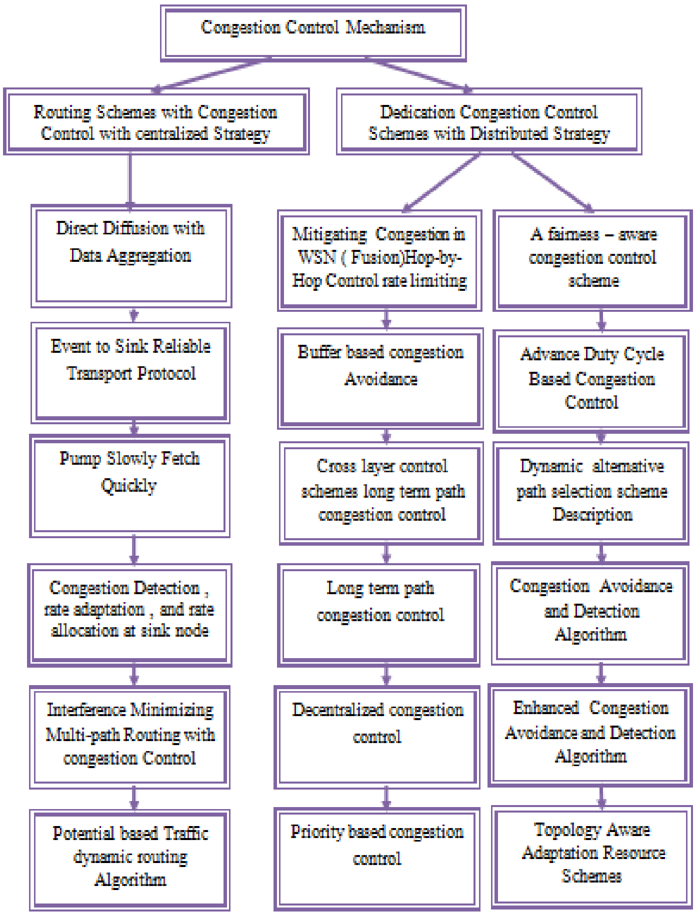


Figure 1: Congestion control schemes

4. The Proposed Approach:

The proposed approach Based on Radial Basis Function Neural Network using Modified Artificial Neural Network Congestion Controller (RBFNNCC) and Shortest path Routing algorithm is divided into two phases as explained below:

4.1. Shortest paths Algorithm phase:

To find as many paths as possible between a pair of source destination nodes. Firstly, let us considered the source node as s ∈ N (where N is the total nodes number) from which it is desired to get the base station node Bs∈ N, through K shortest loop-less paths. These paths form a set of K ingredients, P = {P1, P2, · · · , PK}. The path of the latter set is defined by as follows: PK = hs = v K 1 , vK 2 , · · · , vK i = ti, where v K i is the i-th node of the K-th shortest path. In the next algorithm, which is called shortest path routing (SBR), there is a brief outline of this procedure.[8], [9]

SBR-Algorithm :

Require: A, s, Bs, K.

1: P^1 ← Dijkstra(s, Bs)

2: D ← {P^1} %Set of candidates

```
3: P ← {} %Set of the K shortest paths
4: for k from 2 to K do
5: SP ← the shortest path in D
6: v ← the deviation node (SP, P)
7: P ← P + SP
8: while v 6= Bs do
9: discard all nodes of SP from s to v
10: discard each output link of v which belongs to P
11: SP0 ← Dijkstra(v, Bs)
12: join SP0 and SP from s to Bs
13: D ← D + {SP0}
14: restore all discarded nodes and links
15: v ← successor(v, SP)
16: end while  17: end for
```

In this paper, after distributed the nodes in the sensed area, the above shortest path algorithm is implemented. For each node its state is ON in the network. After that each ON node know its shortest route to the base station. And can use this path to send the sensed information that requested by the base station. The steps of algorithm in this paper are as shown in figure (2):

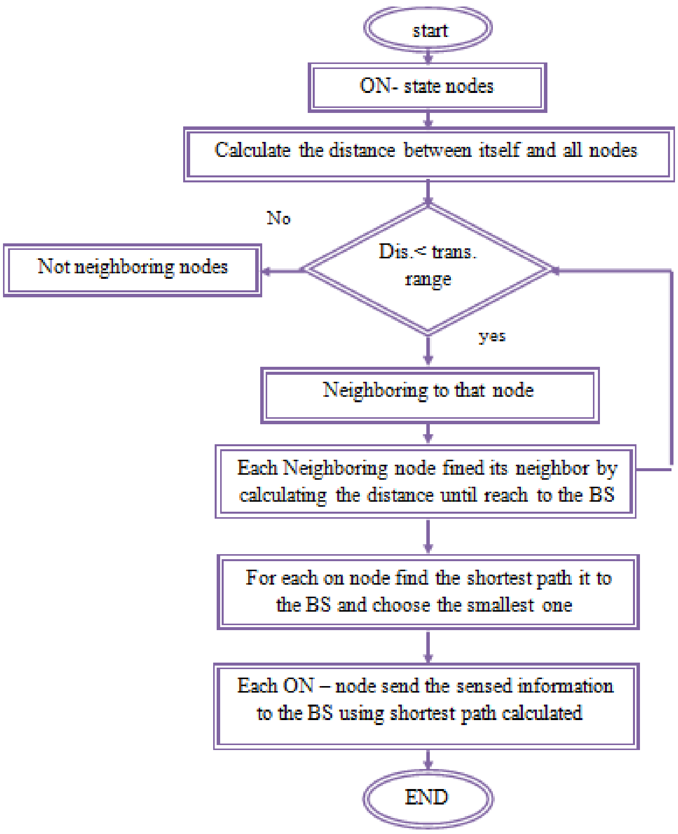


Figure 2: Flowchart for Shortest Path Routing algorithm

4.2. RBFNNCC phase:

The basic idea of the proposed approaches is that:
At the beginning a source node starts to send data with a definable rate through the path. The base station has unit called congestion detection unit. It is activated at reception of every data information. Congestion detection unit has a pointer depending on the state of the that pointer, congestion control operate in order to keep the level of the network traffic at an acceptable value by adjust the data loading rate for each ON-state node. As a result, all the ON nodes have their rates controlled by the base station and prevent the congestion in the memory of the base station. The aim is to create a model to estimate the traffic in the network for the next round time at base station to avoid occurs of congestion on it.
The simple structure of the proposed controller in the base station is shown in Fig. (3).
Where BS is the desired memory occupancy.
Tr (t) is memory occupancy at time instant t.
Rn(t) is the sending rate of ON sensor i.
E(t) is the error.
Error is difference between the desired memory occupancy and actual memory occupancy.

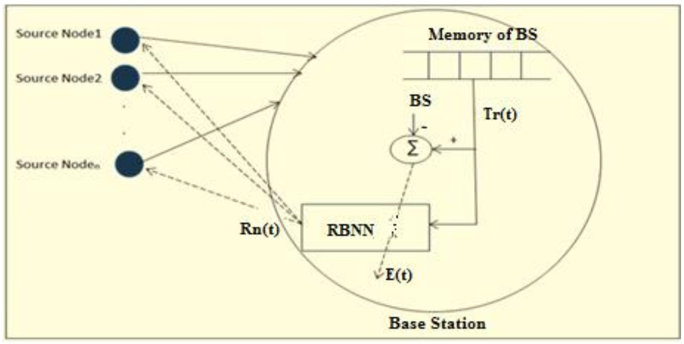


Figure (3). The structure of Base Station controller

The sensors should find appropriate way to the base station node using the shortest path routing. Each sensor before sending its sensed data to the base station it should check its path to evade sending to OFF node.
The controller designed at the base station consists of three major phases:
Congestion Detection Phase (CDPH).
Congestion Monitoring Phase (CMPH).
Congestion Information Phase (CIPH). As shown in figure (4).

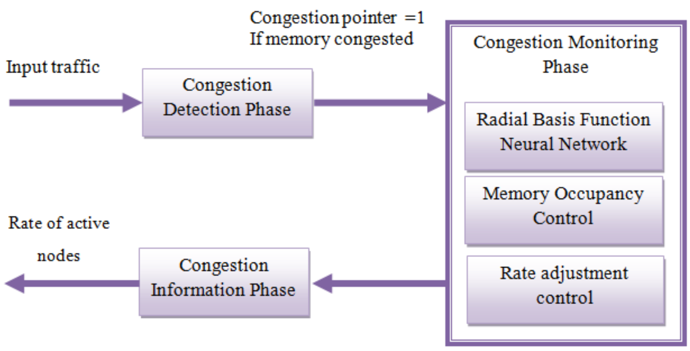


Figure 4: The block diagram of the proposed RBFNNCC

4.2.1 Congestion Detection Phase (CDPH) :

This phase is an important phase in order to control the congestion. Because of the increasing traffic in the network, congestion may occurs, then it will be detected in a timely way by the RBFNNCC in the Base station by calculating Data Loss Ratio DLR and a combination of actual memory occupancy Tr(t). If the threshold of base station memory occupancy be BB max the congestion occurs when the traffic that coming as input to the memory of the base station greater than the size of the memory of the base station. This mean the data lost ratio (DLR) greater than zero, which is represented by the following formula:

DLR=[(send data – received data)/ Total number of data send]*100%

There is a pointer in this phase, this pointer become high state (True) when the congestion occurs. This informs the base station congestion has occurred. After detection of congestion then the designed controller decides either to increase the rate of ON nodes or decrease the rate of ON nodes simply according to the value of congested pointer (CP).

4.2.2 Congestion Monitoring phase (CCPH) :

In this phase a feed-forward Neural Networks used radial basis activation function for hidden neurons with a single hidden layer are called Radial Basis Function (RBF) networks to avoid and reduced the congestion.

4.2.2.1 Radial Basis Function Neural Networks: [10], [11], [12]

There is a different way to construct the radial basis function of neural networks (RBF) has using Multi Layer Perceptron (MLP) networks with a hidden layer of sigmoid unit. The characteristics of (RBF) are discussed as:

- They are constructed as two-layer feed-forward networks and the nodes in a hidden layer of feed forward networks are applied as a set of radial basis functions like Gaussian functions.

- The output nodes are acting as linear summation functions, as in the case of MLP. The training of network is divided into two steps.

(1) The first step: in this step should calculate the weights from the input to hidden layer,
(2) The second step: in this stage calculate the weights from the hidden layer to output layer.

Radial Basis Function Networks (RBFN) are consists from three layers, these layers are: input layer, hidden layer, and output layer.

Hidden units are called radial centers which is representing a set of functions that has an arbitrary basis for the input patterns and these are represented by the array c1, c2… ch as shown in figure(5). To transform from input layer to hidden layer the process is nonlinear but the transformation from hidden layer to output layer is linear process.

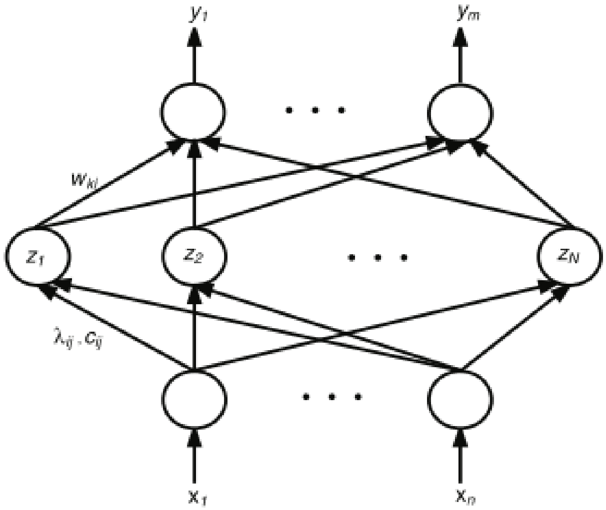


Figure 5: Radial basis function model

In the hidden layer only when the input falls within a small localized region of the input space a radial basis function produces a non-zero response. In the input space each hidden unit has its own receptive field. An input vector x_i which occurs in the open field for center c_j will be activated as c_j and by proper selection of weights, the target output is attained. Here the output of field is related as:

$$\mathbf{Y} = \sum_{j=1}^{h} \emptyset j = \emptyset(\|x - cj\|) \quad (1)$$

Wj: weight of j^{th} center.

The RBFN it is needed a most important selection of the parameters vectors c_i and w_i, i = 1… h to learning. To update the weights and centers of RBFN many techniques are used. These techniques are:

- **Pseudo-Inverse Technique (Off line)**
- **Gradient Descent Learning (On line)**
- **Hybrid Learning (On line)**
- **Pseudo-Inverse Technique**

In this technique one can suppose radial basis functions such as Gaussian functions and the centers are chosen randomly and function is normalized. The standard deviation (width) of the radial function is determined by an ad hoc choice.

- **Gradient Descent Learning**

It is achieved by a gradient descent technique. It is One of the most popular approaches to update c and w, is supervised training by error correcting term . this technique is used in this paper.

- **Hybrid Learning**

In this technique , the radial basis functions using a self-organized manner to relocate their centers while it using supervised learning to update the weights. In both layers, the parameters are updated with the help of gradient descent or a new center is created if the pattern is new.

The network traffic accumulation rate $T_r(t)$ (the number of data arriving to the Base Station) represent the input to the neuron of the NN is assigned. The output of the NN is assigned to estimate the network traffic in next time. As shown in figure 6.

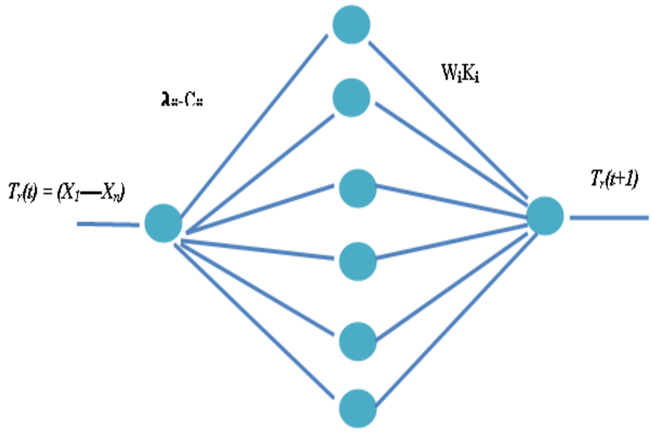


Figure 6: Structure of RBFNNCC control

The following paragraph represent the work of the controller in fig. (6), that is designed at the base station, and the important equations using to find the nets of the neural and the equations to update the weights of the controller.

Tr(t)=sum of active node rate

X (input to the neural)= $T_r(t)$=[$rate_1$ $rate_2$,….]

$net_1=(\lambda_{ij}-C_{ij})'*x$

$fnet_1=sigm(net_1)$

$net_2= W_iK_i*fnet_1$

$fnet_2=purlin(net_2)$

$E(t)=T(t)-MEM(BS)$

Where : MEM(BS): memory size of base station.

$del_1=Ir*error$…………………for weight updating

Where: Ir=0.0001.

$(\lambda_{ij}-C_{ij})new=(\lambda_{ij}-C_{ij})+del_1*fnet_1$ …….. new hidden layer weights

$del_2=lr*(fnet_1*(idd-fnet_1)')*del_1*fnet_1$

Where: idd=[1 1 1 1]

$W_iK_i\ new= W_iK_i+(del2*xx')'$

The block diagram of the Congestion Control Unit is shown in figure (7).

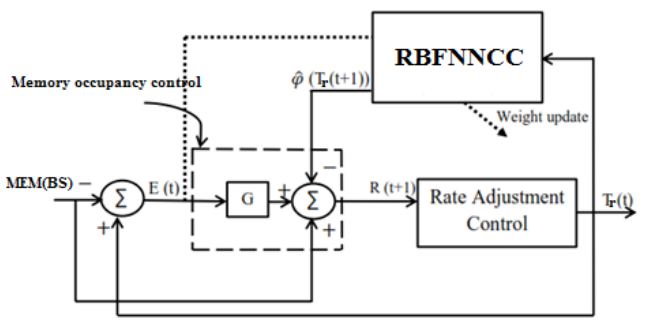


Figure 7: The block diagram of the Congestion Control Unit

From figure(7), one can notice that the error is used to update the weights in NN ,and it can be described in Eq. (2) and the estimated rate is described by Eq. (3)

$$\mathbf{E(t) = T_r(t) - MEM(BS)} \quad (2)$$

$$\mathbf{R(t+1) = [\,MEM(BS) - \varphi\ (T_r(t+1))] + G.E(t)} \quad (3)$$

Where the proportional gain is G, and the estin load of network in next time is R(t+1).

5 . Simulation Results and Network Model :

The simulated network has many sensor nodes distributed in a rectangular area with the following properties:

1. All sensor nodes in the network are stationary and homogenous(has the same characteristics).
2. There are multiple base stations in the network. Nodes can be forward its sensed data to the baste and nearest base stations. And other case with single base station In the network.
3. Nodes do not have location aids such as GPS. Because that they are not location- Aware.
4. The goal of this work is to study the workings of the proposed algorithm with relation to congestion drops. data loss due to collisions are independent design aspects and thus can be abstracted when studying a congestion avoidance and control protocol. Table (1) shows the important information about the designed network.

Table 1: General Information about the Network

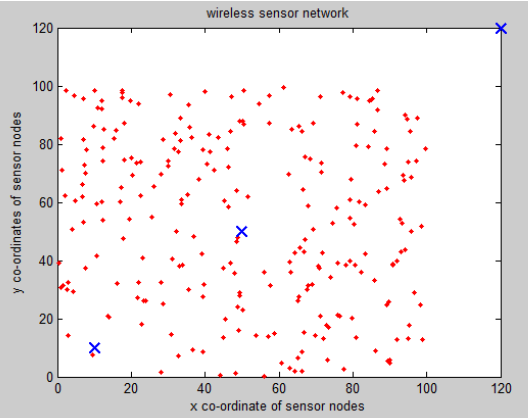
Name	Properties
Area	rectangular
Sensor Devices	250
Sensor-state	ON- OFF
Memory space of base statio	270
emory space of sensor devic	25
Number of Base Station	Case 1: one Case 2: Two
Number of ON – State senso	10-80
Number of rounds	10
Traffic type	No priority
Routing Algorithm	Shortest Path
Rate of data generation	10-40 packet/msec
Channel Bandwidth	**2(Mega bit per msec.)**

In the matlab 250 sensor with three base stations are deployed randomly in a rectangular area as shown in figure (8).

The base station possesses much more computational power, larger memory and is often connected to better energy source than batteries (like: power grid). the base station can be looked as an entry point to the WSN where the primary goal of base station is to gather sensed data from sensor nodes in WSN. Other important objectives, the base station has to accomplish, might be data visualization and analysis. In some cases the base station also handles sensor network routing or node configuration. Furthermore, base station may forward collected data to a remote server application where much wider analysis of data from many WSNs can be performed.

WSN's purpose is to transmit sensed data to the base station. This implies that requests (if particular WSN type supports requests) move from the base station to sensor nodes, but more importantly, sensed data moves in the opposite way (i.e. from sensor nodes to the base station). [13]

at the beginning the shortest path routing algorithm is applied in order that each node find the optimal path to the base station. Then when the node send the sensed data to the base station, the RBFNNCC used to manage the traffic in the network.



X*: *Base stations ***. : Sensor nodes***

Figure 8: Nodes distributed in the Network

At each round time there are the range between (10 – 80) nodes to be in ON – state and the other are OFF. when an event occurred these ON nodes find the shortest path to the base station. The controller at the base station manage the traffic and try to avoid the loss of important information in the sensed data, by prevent the congestion at the memory of base station.

The proposed approach controller tries to manage network traffic by adjusting sources rate fairly and efficiently to avoid congestion at the base station. It adjusts sources rate according to the available capacity of the node memory that the controller implemented on it. The number of sensor nodes that can be ON is between 10 nodes to 80 nodes; moreover, it clarifies that the node remained ON for a specific time according to the existence of the event of that ON node. figure (9) shows the number of ON sensor nodes in the network with and without using controller.

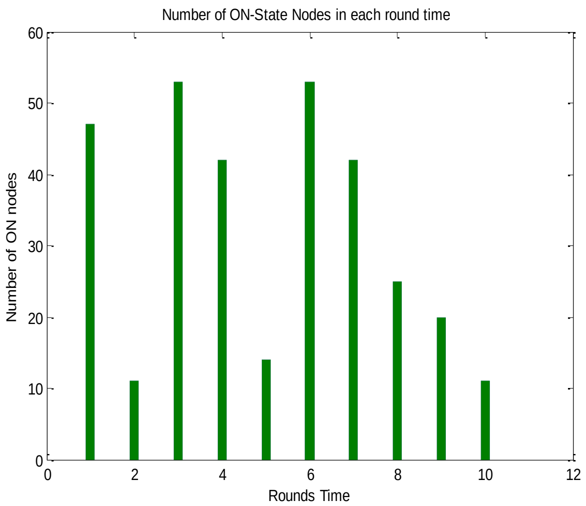


Figure 9: Number of ON-State nodes in each Time

The implementation of the shortest path algorithm in matlab simulation for the second round time, which it is has 14 ON-node, is as shown below to represents the work of the algorithm to choose the shortest path for each onde:

Round time : 2

ON- node : 1

Node 1 sends RREQ to node 4 14 22 23 30 39 40 47 59 62 68 76 80 84 87 88 89 98 102 103 111 123 147 150 153 162 174 175 179 180 182 184 186 191 202 205 206 210 211 213 222 226 228 238 239 240 244 246 250

Each node of source neighbor 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 sends RREQ to its neighbor nodes until reach BS 251

BS 251 sends RREP to node 1 through 251 153 1

ON- node : 2

Node 2 sends RREQ to node 9 22 29 39 42 48 68 82 84 86 87 88 94 96 102 113 119 123 127 131 132 136 141 143 153 162 165 168 174 177 180 183 188 197 200 206 217 220 233 251

BS- 251 sends RREP to node 2

ON- node : 29

Node 29 sends RREQ to node 2 3 15 18 20 39 41 42 49 54 57 63 70 72 82 85 88 91 96 119 123 124 141 143 154 157 165 168 174 180 183 188 189 194 197 198 200 207 216 233 247

Each node of source neighbor 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 sends RREQ to its neighbor nodes until reach BS 251

BS 251 sends RREP to node 29 through 251 2 29

ON- node : 68

Node 68 sends RREQ to node 1 2 4 22 23 30 39 40 42 47 59 62 76 80 84 87 88 89 98 102 103 111 123 147 150 153 162 165 174 175 179 180 182 184 186 202 205 206 210 211 213 222 228 239 240 244 246 250

Each node of source neighbor 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 sends RREQ to its neighbor nodes until reach BS 251

BS node 251 sends RREP to node 68 through 251 206 68

ON- node : 147

Node 147 sends RREQ to node 1 4 22 23 30 35 40 47 59 62 68 76 80 84 87 89 98 101 102 103 117 123 150 153 162 174 175 179 182 184 186 191 202 205 206 210 211 213 222 226 228 239 240 244 246 250

Each node of source neighbor 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 sends RREQ to its neighbor nodes until reach BS 251

BS 251 sends RREP to node 147 through 251 84 147

ON- node : 162

Node 162 sends RREQ to node 1 2 4 9 21 22 23 28 30 37 40 56 59 64 68 84 89 98 102 103 117 123 136 145 147 153 155 182 205 206 210 213 222 228 239 243 246 248 251

BS 251 sends RREP to node 162

ON- node : 190

Node 190 sends RREQ to node 6 13 17 24 26 34 46 52 60 67 105 106 114 115 133 139 140 144 169 195 208 215 219 231 249

Each node of source neighbor 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 sends RREQ to its neighbor nodes until reach BS 251

BS 251 sends RREP to node 190 through 251 112 249 190

ON- node : 195

Node 195 sends RREQ to node 7 13 17 24 34 43 46 52 60 67 83 106 114 115 121 125 133 139 140 144 156 169 185 190 208 215 231 249
Each node of source neighbor 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 sends RREQ to its neighbor nodes until reach BS 251
BS 251 sends RREP to node 195 through 251 167 115 195

<u>ON- node : 196</u>

Node 196 sends RREQ to node 8 21 25 27 31 35 45 50 55 61 65 71 74 76 92 100 101 107 117 126 135 138 142 149 152 170 191 203 205 209 226 229 232
Each node of source neighbor 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 sends RREQ to its neighbor nodes until reach BS 251
BS 251 sends RREP to node 196 through 251 21 196

<u>ON- node : 218</u>

Node 218 sends RREQ to node 5 10 58 73 75 77 81 93 95 97 120 122 125 129 163 181 199 223 236 237 241 242 245
Each node of source neighbor 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 sends RREQ to its neighbor nodes until reach BS 251
BS: 251 sends RREP to node 218 through 251 122 218

<u>ON- node : 243</u>

Node 243 sends RREQ to node 9 10 11 16 22 23 28 33 36 37 44 48 56 58 64 66 79 84 90 94 112 113 116 122 129 130 131 145 148 155 161 162 163 167 172 192 201 206 220 223 225 245 248 251
BS: 251 sends RREP to node 243

6. Performance Metric for Evaluation:

To doubtlessness of the performance of the designed controller, the following metrics are evaluated, and compared with the case of network without using the designed controller.

6.1. Traffic T(t):

The total number of data received to the memory of base station from the nodes that are in ON- state in each round.

$$T_r(t) = \Sigma(\text{Datas of ON nodes})$$

The following figure (10) represent the number of traffic in the memory of the base station that coming from ON nodes with and without RBFNNCC.

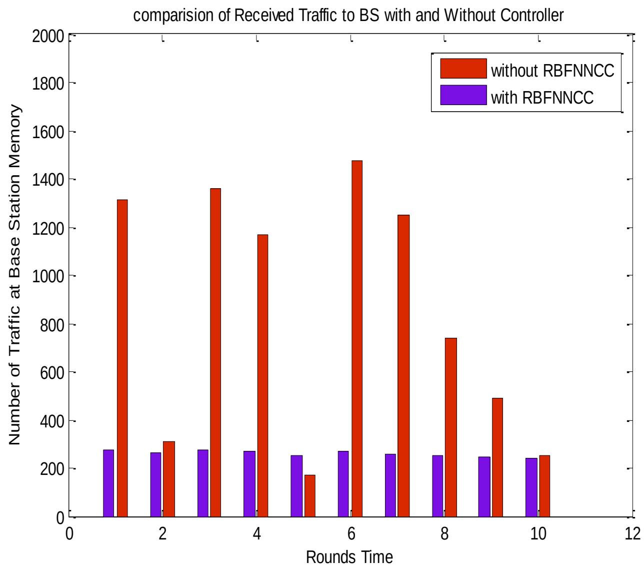


Figure 10: Comparison of Traffic Received At BS with and Without Controller

From the above figure one can see the role of RBFNNCC in managing the traffic that received to the base station in 10 times in fig (9) the base station has congestion (in rounds time 1,2,3,4,6,7,8,9) and losses some of the received data, and other once has no congestion (time 5,10) in the case without using RBFNNCC. But with using RBFNNCC all times has traffic less than the size of the memory of base station (270). Because the Radial Bases Function neural network train the memory of B.S. to receive data in the range of its size not less or more from 270.

6.2. Execution time:

Represent the total time taken by network nodes to distributed in the sensed area then applying shortest path routing protocol phase and RBFNNCC phase, which include (CDPH and CCPH). Because the training of the coming traffic to the base station the case of controller take time more than the case without controller.

Time with RBFNNCC equal to = 256.281308 seconds.
Time without RBFNNCC = 218.944719 seconds.

6.3. Data Lost (DL) and Data lost Ratio(DLR):

- ***Data lost :*** This represents the difference between the space base station memory and the number of the received data from the ON-state nodes to the memory of base station.
- ***Data lost Ratio:***
  DLR= (Number of Data lost in the network) / (Number of data generated by sensing nodes)

Fig. (11) illustrates data lost ratio for the network when implement the RBFNNCC on the network represented by blue line and compare this case with case of not implementing of controller represented by red line in the same figure.

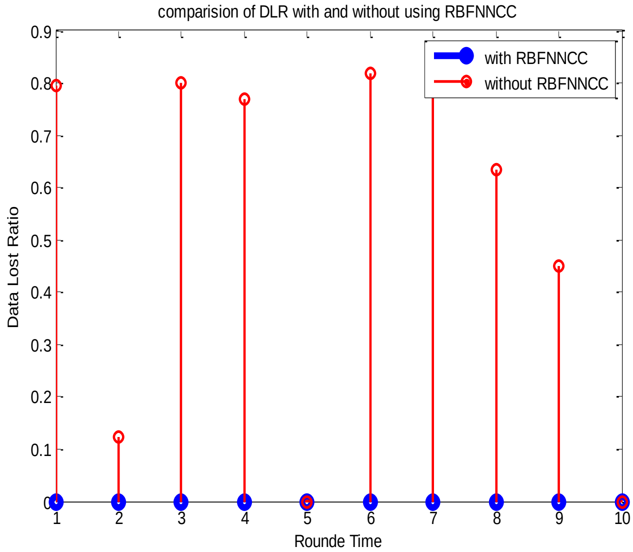


Figure 11: Compression of DLR with and without controller

One can observe from Fig. (11) that the DLR of the network with proposed approaches is better than DLR of the network without controller. This is because the proposed congestion controller decreased sending rate of the active nodes during transmission process in heavy network traffic. Table (2) Contain the number of lost data from each ON node in each round with and without RBFNNCC

Table 2: Number of lost data from each ON node in each round with and without RBFNNCC

Round	ON-Nodes	Traffic without controller	Traffic With controller	DLR without controller	DLR with controller
1	47	1313	269.3724	0.794364	0
2	11	308	263.1574	0.123376	0
3	53	1358	269.5054	0.801178	0
4	42	1170	267.2734	0.769230	0
5	14	169	253.6634	0	0
6	53	1476	267.1374	0.817073	0
7	42	1248	260.0334	0.783653	0
8	25	737	251.0114	0.633649	0
9	20	491	246.6834	0.450101	0
10	11	254	243.4094	0	0

For each round in the case of RBFNNCC there is no data lost from the memory of the base station, this is because that it is training by the RBFNN to accept data in range less than 270 (space of base station memory).

6.4. Memory Utilization of the Base station (MEMU):

The memory utilization is defined as the ratio of the data which is occupied at the B.S's input memory with the memory length as described in Eq. (5).

MEMU=(Total number of data sensed by nodes)/(space of memory o base station) (5)

Table (2) represent the MEMU without and without controller , MEMU must be approximately equal to one. MEMU is very important in the traffic at the Base Station. If its value approximately equal to one, then the B.S. has no congestion. This is because the RBFNNCC, which decrease the redundant data arrived to the B.S., and decrease the congestion that may be occurs. Table (3) contain the values of MEMU with and without using controller.

Table 3: The values of MEMU with and without controller

Round	MEMU without controller	MEMU with controller
1	4.862962	0.990268
2	1.140740	0.974657
3	5.029629	0.998168
4	4.333333	0.989901
5	0.625925	0.939494
6	5.466666	0.989397
7	4.622222	0.963086
8	2.729629	0.929671
9	1.818518	0.913672
10	0.940740	0.907590

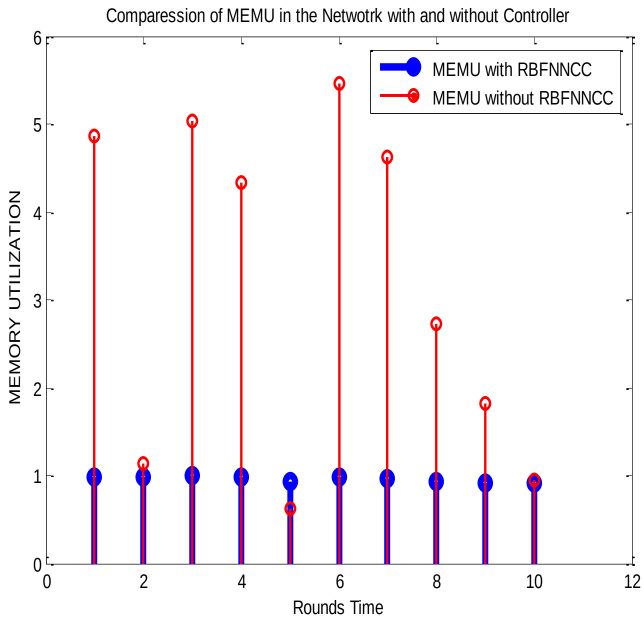


Figure 12: Comparison of MEMU in the Network with and without Controller

From figure(12) that represent MEMU comparison between the network with and without controller. One can notice that, the case of network has MEMU values greater than one and this is bad value. Because the congestion in the memory of base station. But the case of network with RBFNNCC all the MEMU values approximately equal to one, because the role of controller in solving congestion and managing the traffic arrived at the base station. From the above figure one can see that the MEMU without controller has cases greater than one this means that the base station lost some of the coming data from the ON nodes in some rounds because the incoming data greater than the memory space. But in the case of using RBFNNCC all the MEMU is approximately equal to one, because of the role of controller that adjust the traffic coming to base station to value equal approximately to 270 (B.S memory) by applying RBFNN and updating the weights until the traffic training.

7. CONCLUSION

Congestion problem in wireless sensor network and how can manage the traffic in the network was studied in this paper. Shortest path routing and multilayer perceptron feedforward radial bases function neural network was used to model the network. The network was trained using actual traffic data as an input to the neural. The trained network was simulated by using MATLAB with new input data to the base station, the results show that the behavior of the developed model is success in solving congestion and manage the traffic occurs in the network. With availability of relevant historical traffic data, radial basis neural networks can model the behavior of network to predict the occurrence of network congestion and manage the traffic.

7. References:

[1] Mirza Waseem Hussain, Sanjay Jamwal and Majid Zaman," Congestion Control Techniques in a Computer Network: A Survey",*International Journal of Computer Applications (0975 – 8887) Volume 111 – No 2, February 2015*.

[2] Rajender kumar, Ravinder Khanna, Pankaj kumar and Surender," A proposed Work on node clustering & object tracking processes of BFOA in WSN ", IJCSC Volume 4 • Number 2 September 2013 pp. 207-212 ISSN-0973-7391.

[3] Narendhar, S., Amudha, T., (2012), “A Hybrid Bacterial Foraging Algorithm for Solving Job Shop Scheduling Problems,” International Journal of Programming Languages and Applications (IJPLA).

[4] Razieh Golgiri and Reza Javidan," TMCC: An Optimal Mechanism for Congestion Control in Wireless Sensor Networks", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 5, 2016.

[5] R. Abinaya and S. Kamakshi, " Improving QOS using Artificial Neural Networks in Wireless Sensor Networks", Indian Journal of Science and Technology, Vol 8(12), DOI: 10.17485/ijst/2015/v8i12/63283, June 2015 ISSN (Print) : 0974-6846 ISSN (Online) : 0974-5645.

[6] Sayd afsar shah, babar nizar and imran ali khan," Congestion control algorithms in wireless sensor networks: Trends and opportunities jornal of king Saudi university – computer and information science volume 29, Issue 3, July 2017, Pages 236-245.

[7] Ee and Bajcsy, " Congestion control and fairness for many-to-one routing in WSN", Proceedings of the Proc. ACM Sensys’04 (2004).

[8] Wilton Henao-Mazo and Angel Bravo-Santos, " Finding Diverse Shortest Paths for the Routing Task in Wireless Sensor Networks", ICSNC 2012 : The Seventh International Conference on Systems and Networks Communications.

[9] W. Hoffman and R. Pavley,“A method for the solution of the Nth best path problem”. Journal of the ACM, 6: pp. 506-514, 1959.

[10] Navpreet Kaur, Kuldeep Singh and Hardeep Kaur, " Radial Basis Function (RBF) Based RoutingOptimization for Wireless Sensor Network", International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 6, June 2014.

[11] Mark A. Kon, Leszek Plaskota, "Neural Networks and Radial Basis Functions, and Complexity", Research partially supported by the National Science Foundation, Air Force Office of Scientific Research, and U.S. Fulbright Commission.

[12] Navpreet Kaur, Kuldeep Singh and Hardeep Kaur, "Back Propagation Neural Network (BPNN) Based Routing Optimization For Wireless Sensor Network", International multi track conference on sciences, engineering and technical innovations, Vol 1, june 2014.

[13] Michal Michal´ık, " Base station for Wireless sensor network", DIPLOMA THESIS, 2013.