Wireless Sensor Network Congestion Control Based on Adaptive Neuro-Fuzzy Inference System: IParking application using NODEMCU

***Takoua Mekni[1†]*, Kaouther Ibn Taarit[1††], Moufida Ksouri[1†]**

[1†]University of Tunis El Manar, National Engineering School of Tunis, LR11ES20 Analysis, Conception and Control of Systems Laboratory (LACS), Tunisia
[†]University of Monastir, National Engineering School of Monastir,

Summary
Wireless sensor network (WSN) is a growing technology used to perform different tasks. Due to their source limitation such as lack of memory, computation resource and power, WSNs are susceptible to faults and to be congestion which is one of the its main problems. Congestion causes Quality of Service (QoS) degradation due to delay, energy consumption and packets loss which decreases the network lifetime.
In this paper, an adaptive neuro-fuzzy inference system (ANFIS) congestion control protocol is proposed and tested on an IParking that we developed. The congestion is detected in the sink node and controlled through it by adjusting the sending rate in the source nodes. Local parameters were used to estimate congestion like participants, traffic rate and buffer occupancy. We used a real application to evaluate our protocol: parking service that is easy to implement in existing parking lots, low cost and provided with an efficient sensor to track vehicles. Our results prove the efficiency and the reliability of our protocol.

Key words:
Wireless Sensor Networks (WSNs); Congestion; Adaptive Neuro-Fuzzy Inference System (ANFIS); Control; NODEMCU; Arduino; MATLAB; IParking

1. Introduction

Internet of things got in the last few years a lot of intention due to its potential for embedded applications. Those applications are based on the use of wireless sensor network which are the one of the most important elements of IoT and was influenced by its advancement[1].

Wireless sensor networks are gatherings of small size, low cost, self-configured and battery powered sensors and are able to cope with difficult environmental conditions. WSN is a group of interconnected autonomous sensors that communicate wirelessly to monitor environmental and physical conditions such as sound, temperature, humidity, pressure etc. These networks have been deployed for different fields like industrial, military, healthcare and environmental monitoring [2].

Due to their processing, memory and power limitations, a WSN faces a lot of issues. They are deployed in harsh environments and have been under busy traffic collecting measurements in a short time and sending it to the sink node therefore Data is more exposed to delays and packet loss thus congestion occurs. WSNs are more exposed to congestion than wired networks because of the topologies bandwidth, important delays, interference and unbalanced wireless links.

Congestion is a common issue for packet exchanging networks. It appears when active sources nodes are sending mass data that surpasses the capacity of the network processing when events are detected. The intermediate routes are surcharged and the congestion will be passed to the nodes [3]. There exist two types of congestion: the first type is node level congestion which occurs as a consequence of overflowed buffers in a particular node and the second one is link level congestion caused by packet collisions.

The main reasons for congestion occurs are: the buffer overflow which take place when the available buffer can't handle the incoming packets, the many-to-one communication between various sources and sink creating traffic jam around the sink, the channel contention appears between multiple packets of a flow and different flow, and the packet collisions that causes packet drops [4].

Congestion leads to QoS degradation due to the increased energy consumption, end to end delay and packet loss. It decreases the reliability of data transmissions and detection. When it occurs in a node, it may be diffused to the network and degrade its whole performance. In some applications that require large data-rate, congestion detection and control are vital to maintain high network performance.

Manuscript received December 5, 2020.
Manuscript revised December 20, 2020.
https://doi.org/10.22937/IJCSNS.2020.20.12.13

The main contribution of this paper is to control congestion in sink node using an Adaptive Neuro-Fuzzy Inference System (ANFIS). Our method aims to provide the best and fastest congestion detection thanks to the fusion of two artificial intelligence methods. The Neuro-fuzzy method brings out the main assets of each technique and overcomes the weaknesses. In the case of the Fuzzy logic method, the main problems are its loyalty to the rule base and inability of mitigation due to the topologies changes. For neural networks, setting the number of layers and neurons may be difficult and takes a massive amount of time.

With the constant rise of car numbers, parking lots are deployed everywhere and we are in need for more parking with smart systems.

To test the performance of our protocol, we opted for a low cost and low power consumption IParking based on NODEMCU ESP8266 module integrated for internet communication.

The remainder of this paper is structured as follows. Section 2 presents the most relevant work. Section 3 discusses the problem statement. The proposed scheme for congestion control is described in detail in Section 4. Experiments and results are presented in Section 5 and the last Section is devoted to conclusions.

2. Related Works

In this section, we are going to briefly survey some of the most used and recent congestion detection techniques and control protocols. The difference between them can be in the way of detecting, correcting or avoiding congestion.

CODA [5] is one of the most known congestion control techniques. CODA is the abbreviation of COngestion Detection and Avoidance. It's an efficient control technique that can solve congestion from the sensor to the sink i.e upstream direction. It's a receiver based detection technique that depends on the buffer occupancy status in the present and in the past. When the threshold value is exceeded, the congestion is detected and the neighbor's nodes will be notified. This node will reduce the flow by backpressure mechanism. CODA bases its congestion detection technique on wireless channel loading and queue length. For the control part, it uses rate adjustment technique AIMD and regulates the rate by hop-by-hop and end-to-end controls.

Event-to-Sink Reliable transport (ESRT) [6] is a reliable congestion control protocol that uses the minimum of energy expenditure. The main purpose of ESRT is to overcome CODA disadvantages. This method is based on event reliability which is the number of packets received at the sink decision interval and the sensor reporting frequency regulates end-to-end delivery service. ESRT makes sure that the observed event reliability is greater than the desired value to avoid congestion.

The local buffer level is used to detect congestion in the nod and a notification bit in the packet header is added when congestion is detected.

Artificial intelligence based techniques were used to control congestion. Some of these methods are exposed in the rest of the related work.

Fuzzy Based Adaptive Congestion Control (FBACC) [7] estimates congestion using a fuzzy logic system. Congestion level is determined by exploiting three parameters: participants, buffer occupancy and traffic rate. Then, the data rate is adapted to reduce the packet loss. The authors compare their method to FLCE (Fuzzy Logic-based Congestion Estimation for QoS), ESRT (Event-to-Sink Reliable transport), and CCSFL (Congestion Control Scheme based on Fuzzy Logic) in term of performance and proves the precision and the efficacy of FBACC to reduce packet loss and reduce energy consumption by minimize the packet retransmission.

[8] uses Fuzzy Inference System (FIS) to control congestion in WSN. Congestion is estimated using buffer occupancy and a congestion index which is the ratio of service time over interarrival time. These two parameters are fuzzy logic inputs for each node. In case of congestion, the rate will be adjusted according to the decision values.

In [9], the authors propose a control method of congestion effects in WSNs with mobile sink nodes based on Wavelet Neural Networks (WNN). This method uses the flexibility of the sink node to reduce packet loss and enhance the QoS since node mobility is an effective way to reduce communication hops between nodes.

The congestion is estimated by the sink node using buffer occupancy.

When 90% of the buffer is occupied, the congestion control is activated, the congestion is detected using a neural network and the data rate is adjusted in the cluster head nodes.

Fuzzy logic Control [10] introduces two fuzzy based algorithms and it's composed of 3 units: detection unit (CDU), notification unit (CNU), rate adjustment unit (RAU). The first unit estimates the congestion level by the use of buffer capacity and congestion index. If the congestion is detected, CNU is activated and an implicit signal is sent to the rate adjustment unit that will control the traffic rate by the use of another fuzzy system.

Neural Network Based Congestion Detection protocol [11] is the first to introduce neural networks to the congestion detection systems. The congestion is estimated in the sink node using the traffic rate, buffer occupancy and number of participants. It uses network simulator NS2 to generate parameters, AWK scripts to extract them and MATLAB software to train the neural network.

Neural Network Congestion Detection Protocol (NNCD) [12] estimates congestion level in each intermittent node in the network by the use of local information traffic rate, the number of participants and the buffer occupancy as inputs for neural network and identifying malicious nodes in network.

Fuzzy Congestion Control Protocol Based on Active Queue Management in Wireless Sensor Networks [13] introduces a hybrid protocol using fuzzy logic and PID controller (fuzzy PID) for congestion control. The presented protocol is composed of three units: detection unit, implicit notification unit and control unit.

The detection unit calculates the average of queue length (qavg) of each node which is very similar to RED protocol for queue management. Due to the lack of linear relationship between packet drop rate and aforementioned parameters, a fuzzy proportional integral derivative, that combines RED method, defines two thresholds for packet drop rate calculation.

It uses two inputs for the fuzzy logic system: conventional error and the error between queue length and target queue length. The output is the dropping probability.

After the calculation phase, a notification signal is sent to the intermittent nodes and finally the control unit regulates the rate of the nodes with an appropriate one using fuzzy logic.

3. Congestion Control

3.1 Proposed technique

A neuro-fuzzy congestion control protocol was developed in which the main advantages of neural networks and fuzzy logic are used. The ANFIS is the association of neural networks and fuzzy logic. The used inputs in our protocol are Traffic rate, Buffer occupancy and Participants to better estimate the congestion occurrence and level. The main purpose behind ANFISCC is to decrease packet drop in the sink node. This protocol is developed for applications with low powered and limited memory. The control protocol is composed of three parts: detection unit, notification unit and rate adjacent unit.

3.2 Congestion detection

3.2.1 Input Selection

Through time, different parameters were used for congestion detection such as, Packet Inter-Arrival time, Packet Rate, Packet Service time, Buffer Occupancy, Buffer Size, Channel Load, Queue Length and Packet Delivery Ratio.

Usually, WSN detection protocols use buffer occupancy for estimation. But under busty traffic, one parameter is not enough therefore several protocols add more parameters to guarantee protocol reliability. For our protocol, three variables were used and they were chosen due to their performance efficiency for congestion detection with artificial intelligence methods.

These variables are:

Traffic rate: the number of packets received by sink nodes per second to determine traffic load.

Buffer occupancy: the number of packets addressed per second by the number of packets reached the sink node per unit time.

Participants: the number of active nodes at every second. When it increases, congestion also becomes important.

These parameters are the inputs of ANFIS to calculate an output that represents the congestion level and compared to a set of defined thresholds. If the congestion occurs, the sink will detect it and propagate an external notification which is a specified control packet to notify the source and reduce the traffic rate.

3.2.2 ANFIS Based Design and Structure

Adaptive Neuro-Fuzzy Inference Systems is a data learning technique proposed by Takagi Sugeno and Kang [14] using a fuzzy inference model that transforms a known input to a target output. It combines and benefits two of the most known machine learning techniques (neural network and fuzzy logic) into one technique. It applies neural network learning to fuzzy inference systems (FIS) parameters [15].

These techniques provide a fuzzy modelling to learn information about a given data set to compute the best membership parameters of fuzzy inference systems to pursue the given input /output data.

It is composed of five components: fuzzy sets, fuzzifier, fuzzy rules, inference engine and defuzzifier. It involves membership function, fuzzy logic operators and if-then rules [16].

ANFIS offers numerous advantages due to neuro-fuzzy association [17]:

- It doesn't require human expertise and is easy to develop and implement.
- Offers fast and precise learning.
- It uses If-Then rules to express the behavior.
- It provides a large selection of membership functions to use, exceptional generalizationabilities, and outstanding description facilities by fuzzy rules.

ANFIS architecture is composed of [18]:

Layer 1: Fuzzification layer, each input is assigned with membership level (full, partial or none).

Layer 2: Nodes in this layer are fixed and their outputs are the product of all incoming signals.

Layer 3:It calculates the ratio of individual rules firing strength to the sum of all rules.

Layer 4: Defuzzification layer where the individual outputs are calculated from the inferring of rules.

Layer 5: Output layer where the sum of all outputs from the previous layer is calculated.

ANFIS uses a hybrid learning algorithm to train data that combines least-square method (forward pass) and gradient descent method (backward pass). The least-squares method optimizes consequent parameters to guarantee fast training and gradient descent changes the membership function that engenders the basic functions for the least-squares method. For that, ANFIS can give satisfactory results after one epoch of training.

For this protocol, we used a hybrid algorithm as a learning algorithm. This algorithm can solve the problem of search space, optimize consequent parameters to have the best output and decrease the error between the maintained and desired output.

After a various experiences and performance evaluation, the architecture deployed has the following specifications: Number of nodes: 78, Number of linear parameters: 27, Number of nonlinear parameters: 35, Total number of parameters: 62, Number of training data pairs: 200, Number of checking data pairs: 200, Number of fuzzy rules: 27.

If the congestion level is high, which indicates the occurrence of congestion, the sink node adjusts the rate of sources for some time to reduce congestion.

3.2.3. Traffic Adaptation

The traffic is adapted after the congestion is detected. There are three levels of congestion which are determined by summing the inputs. The levels are defined to get optimal results:

- *Low (summation =< 1):* in this case the node isn't active, the congestion is low and traffic control is not needed yet.
- *Medium (1< summation =< 3):* the packet loss and traffic rate are average and the congestion is acceptable, there is no need to control the rate flow.
- High (summation > 3): the packet loss is high therefore the congestion is high, the congestion control is activated and the rate flow for source nodes is decreased.

3.3. Congestion Notification

When congestion occurs and is detected, the source nodes get a notification signal. There are two types of notification signal: Explicit Congestion Notification (ECN) in which the notification is a separate packet from the data packets: it has its own packet and an Implicit Congestion Notification (ICN) in which the signal is sent with the data packet. In this protocol, we use explicit notification in which the sink node notifies the entire source to nodes in the network to change the rate.

3.4. Rate Adjustment

Once the notification signal is received by source nodes; the transmission rate is adjusted depending on the congestion level. The new sending rate is obtained by varying the delay between the packets. If the level congestion is low, the execution time is maintained. Otherwise the data rate is decreased until the congestion level is low again.

4. Proposed IParking

WSN is a set of cooperative nodes forming a network. Each node is composed of a processor (multicontrollers), memory and transmitter-receiver module (radio or wifi), a power source (solar cells and batteries) and one or many sensors. WSN is deployed in an area to perform a specific task.

For our model, we deployed 4 source nodes to detect the presence of vehicles and send the data via WIFI-Router to the sink which collects the data then sends it to the database and processes it through a serial connection to MATLAB. The database is used in the developed web API. This application suitable for all mobile devices helps the user to park his vehicle by assigning the nearest empty place available and informs him when the parking is full.

The main purpose of this WEB API is to prevent the loss of time for the users. Figure 1 presents the system developed in this paper.

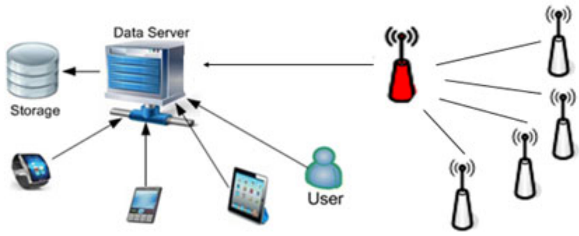

Fig.1 System developed.

4.1. Source node (sensor node)

One of the main parts of WSN is the source node which can be a transmitter and receiver. In our application, sensor nodes are composed of NodeMCU as microcontroller, batteries as a power module and PIR as a presence detector sensor.

Passive Infrared Sensor or Pyroelectric Infrared Radial Sensor is a sensor that detects the changes in a certain distance through infrared light and gives a signal as output. It can detect any object such as human beings or animals in a specific range. It’s low cost, small size, user-friendly, low powered and reliable sensor. Its sensitivity range is up to 6 meters and less than 120° detection range. The Delay time can be adjusted from 3 to 5 min.

NodeMCU is a development board targeted for IoT based Applications. Its firmware runs on the ESP8266 Wi-Fi with TensilicaXtensa 32-bit microprocessor and its hardware is based on the ESP-12 module.

It supports RTOS and operates with an adjustable clock frequency. NodeMCU has 4MB of Flash memory and 128 KB RAM to store data and programs. It’s a small size board with Wi-Fi and Deep Sleep Operating features. It supports SPI, UART and I2C interface. It can be powered using Micro USB jack and External Pin Supply.

The source node is powered by a battery which is linked to NodeMCU and the PIR sensor is powered through NodeMCU with 3.3V of power. The source node built and connected with the sensor is shown in Figures 2 and 3.

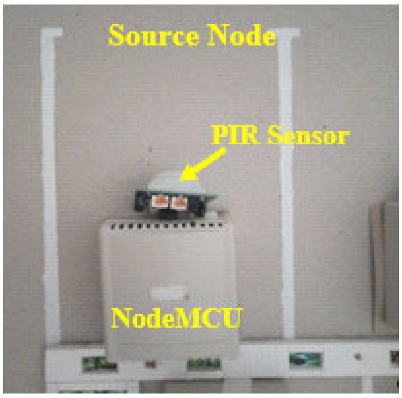

Fig. 2 Built of sensor node.

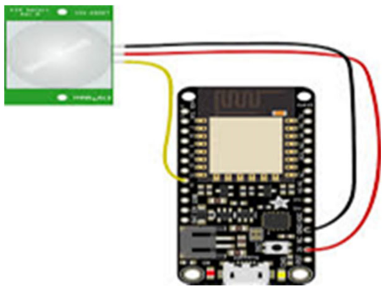

Fig. 3 Connection o NodeMCU with PIR sensor

4.2. Sink node

The sink node is the node that receives data from all the source nodes, store the data and generate the developed web application. It's formed by NodeMCU connected to the PC through a USB cable. The NodeMCU is the server in this case and can send data to the source node. It should be noted that we use UDP protocol to exchange packets in Wireless Sensor Network.

Figure 4 represents the WSN with 4 source nodes and a sink.

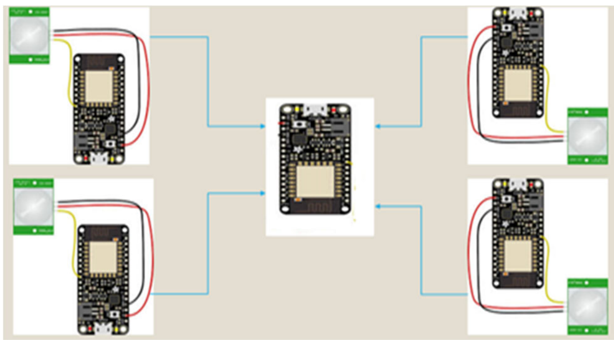

Fig. 4 Wireless Sensor Network.

Figure 5 shows the IParking prototype that we create to test it.

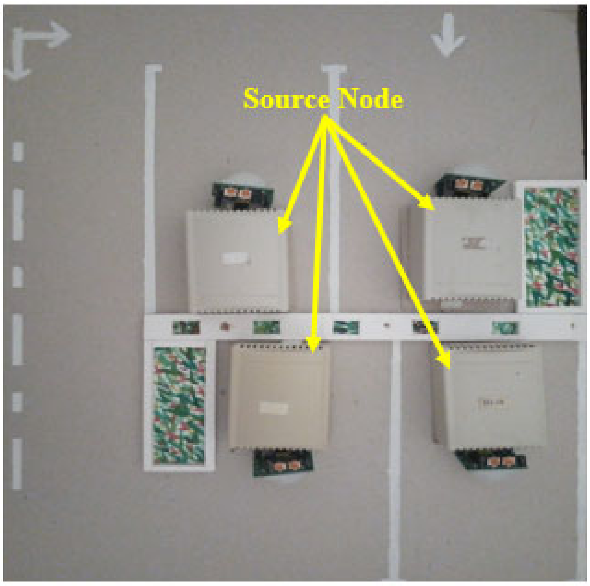

Fig.5 IParking prototype

4.3. WEB APPLICATION

To find a place in the parking via the internet a Web application called "IParking" has been developed with the PHP programming language. It's simple and easy to use application. IParking is a very convenient application and intends to give the user easy access regardless of the location of the device. As already mentioned, the application is suitable for any smart device. To use this application, we don't need to log on just to be in the parking space. The web application gadded the user to the nearest space available. It gives a real time state of the parking occupancy as shown in Figures 6, 7and 8.

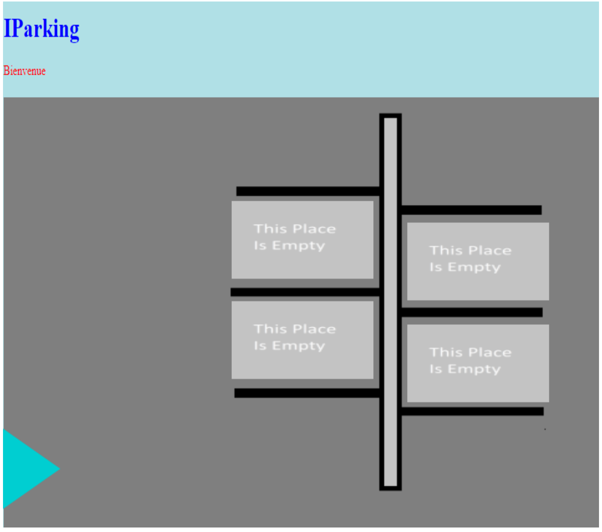

Fig.6 IParking Web page when empty

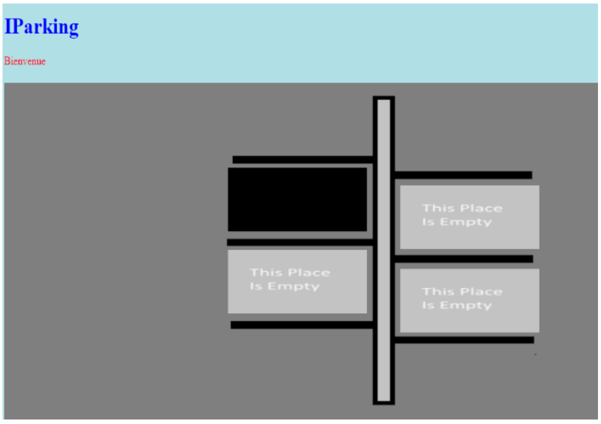

Fig.7 IParking Web page with one car

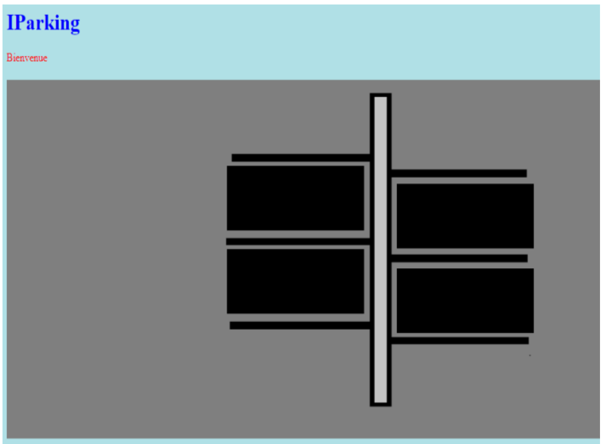

Fig.8 IParking Web page when full

5. Implementation and Results

The proposed Congestion detection algorithm is implemented using MATLAB software. The IParking which is the application we developed is composed of 4 source nodes and a sink node that sends and receives data through serial communication with MATLAB. The proposed ANFIS will detect congestion each minute and inform the sink node to decrease the traffic rate for the most active sources.

For the purpose of evaluating our protocol, we increased the number of data sent to the sink. The size of packets is 20 bytes and the bacon message is 5 bytes.

The data traffic in this application can be described in the following Figure 9.

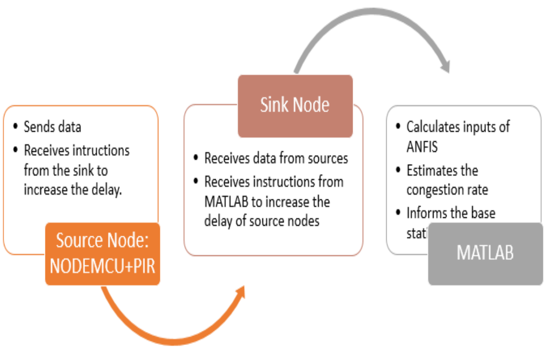

Fig.9 Packet exchange

The sink node sends all the received data to MATLAB through serial communication. This one calculates the input for the ANFIS. When congestion is detected, MATLAB informs the sink of its occurrence which increases the delay between the packets.

To evaluate our protocol, there exist various parameters to estimate its reliability such as Packet Loss Ratio (PLR), Packet Delivery Ratio (PDR), Throughput, End-to-End Delay. These parameters are influenced by packet size, transmission range and node number.

In this case, the following parameters are used [19]:

- **Packet Delivery Ratio:** one of the most important evaluation parameters. It's the number of received packets by the number of packets sent by the source nodes shown in Eq. 1:

$$PDR = \frac{\sum(n\,\mathrm{Re}\,cievedpacket)}{\sum(nSendpacket)} \quad (1)$$

where:

- nReceivedpacket = total number of received packet by destination node;
- nSendpacket= Total number of send packet by all source nodes.

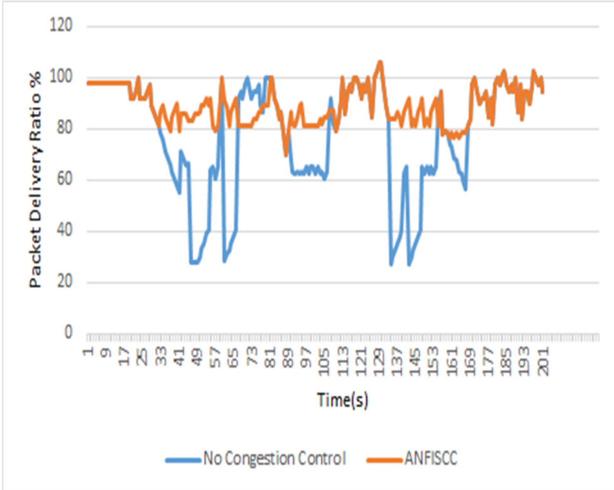

Fig.10 Packet Delivery Ratio

Figure 10 shows the packet delivery ratio variation in time. The performance of our proposed method ANFISCC is proved through PDR which remains constant. The minimum of PDR value is 70% which is acceptable for real application using UDP protocol. The graph does not vary as when no congestion control is applied.

- **Packet Loss Ratio:** the number of packets that didn't reach the destination by the number of packets sent by source node. Often; this ratio will be given as a percentage. It can be shown by the following Eq. 2:

$$PLR_{\%} = \frac{nGeneratedpacket - n\,\mathrm{Re}\,cievedpacket}{nGeneratedpacket} \times 100\,\% \quad (2)$$

where:

- nGeneratedpacket= number of generated packet;
- nReceivedpacket = number of received packet.

Figures 11 and 12 show the variation of the packet loss ratio over time (in second). PRL is considerably less for ANFISCC than the no congestion protocol applied. The maximum packet loss is 20% which is considerably acceptable.

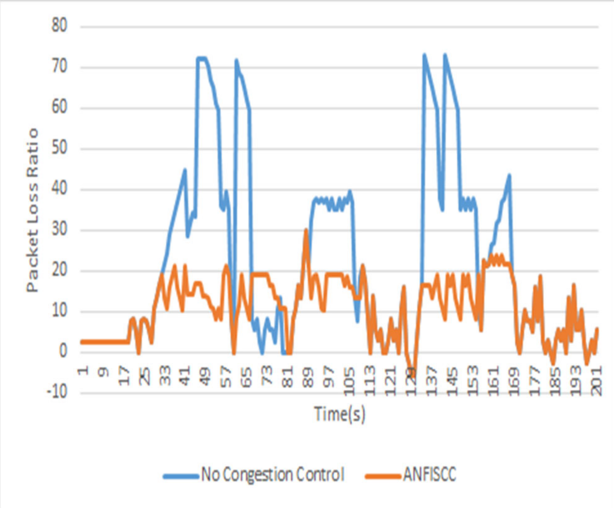

Fig.11 Packet loss Ratio%

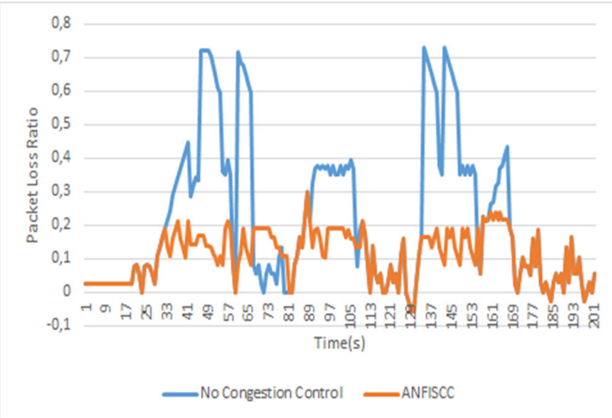

Fig.12 Packet loss Ratio

- **Throughput:** the number of successfully received packets multiplied by the average packet size by total sending time shown in Eq. 3.

$$Throughput = \frac{recievedsize}{stopT - startT} \times \frac{8}{1000} \quad (3)$$

with:

- recievedsize = received packet size;
- stopT= Stop time simulation;
- startT=Start time simulation.

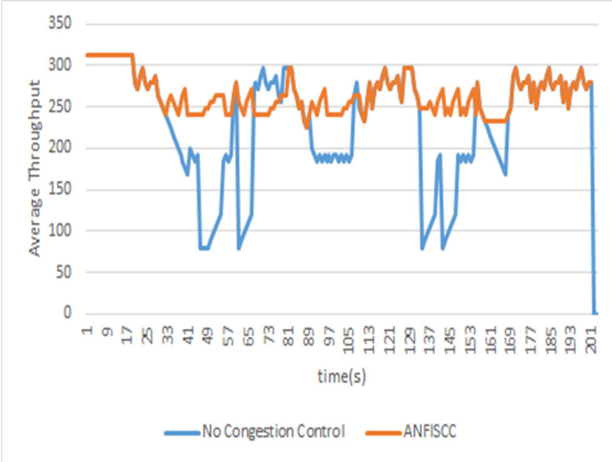

Fig.13 Throughput

Figure 13 shows the throughput variation in terms of time. It may be observed that when ANFISCC is applied, the throughput is or will be constant.

In general, for efficient transmission, the throughput should be high. We can notice from the graph that the throughput is maintained constant through congestion.

Regarding the packet delivery ratio, it is influenced by the number of packets received. For better performance, PDR should be high and maintained constant through time. If PDR is maximized, it shows that all packets transmitted are received. The PDR of the proposed method is maintained higher than the PDR of no protocol applied.

6. Conclusion

In this paper, Adaptive Neuro-Fuzzy Inference System congestion control was proposed to estimate congestion in sink nodes using three parameters as input for MATLAB toolbox namely buffer occupancy, participants and traffic rate. This protocol is found to be able detect congestion correctly whenever there is packet drop and inform the sink node to control it by adapting the transmission rate in the source node. In future work, in order to show its efficiency, a comparative study of our proposed method with existing congestion control techniques we will be considered.

References

[1] Arshad, A., Alshmrany,S. Health Monitoring and Management System by Using Wireless Sensor Network and Internet of Things (IoT). *IJCSNS*, VOL.*19*(12), 179-184 (2019).

[2] Akyildiz, IF., Su, W., Sankarasubramaniam, Y., Cayirci, E. Wireless Sensor Networks: a survey. Computer Networks, vol. 38(4), pp. 393-422 (2002). DOI: 10.1016/S1389-1286(01)00302-4

[3] Wang, C., Li, B., Sohraby, K., Daneshmand, M., Hu, Y. Upstream congestion control in wireless sensor networks through cross-layer optimization. *IEEE Journal on selected areas in Communications*, *25*(4), 786-795 (2007). DOI: 10.1109/JSAC.2007.070514

[4] Kim, J. M., Park, S. H., Han, Y. J., Chung, T. M. CHEF. Cluster head election mechanism using fuzzy logic in wireless sensor networks. In *2008 10th International Conference on Advanced Communication Technology*; vol. 1, pp. 654-659, ICACT2008 (2008, February). DOI: 10.1109/ICACT.2008.4493846

[5] Chang, W. R., Lin, H. T., & Cheng, Z. Z. CODA: A continuous object detection and tracking algorithm for wireless ad hoc sensor networks. In *2008 5th IEEE Consumer Communications and Networking Conference*, pp. 168-174 (2008, January). DOI: 10.1109/ccnc08.2007.44

[6] Sankarasubramaniam, Y., Akan, Ö. B., Akyildiz, I. F. ESRT: Event-to-sink reliable transport in wireless sensor networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pp. 177-188, (2003, June). DOI: 10.1145/778415.778437

[7] Jaiswal, S., Yadav, A. Fuzzy based adaptive congestion control in wireless sensor networks. In *2013 Sixth International Conference on Contemporary Computing (IC3)*, pp. 433-438 (2013, August). DOI: 10.1109/IC3.2013.6612234

[8] Wei, J., Fan, B., & Sun, Y. A congestion control scheme based on fuzzy logic for wireless sensor networks. In *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 501-504 (2012, May). DOI: 10.1109/FSKD.2012.6234353

[9] Al-Jamali, N. A. S., & Al-Tahan, N. R. M. A. Control of Congestion Effects in Wireless Sensor Network with Mobile Sink Node Based on Wavelet Neural Network. *Association of Arab Universities Journal of Engineering Sciences*, *25*(1), 183-194 (2018). DOI: 10.13140/RG.2.2.29221.60644

[10] Ghanavati, S., Abawajy, J., & Izadi, D. A fuzzy technique to control congestion in WSN. In *The 2013 International Joint Conference on Neural Networks IJCNN*, pp. 1-5 (2013, August). DOI: 10.1109/I2CT.2014.7092259

[11] Singhal, P., Yadav, A. Congestion detection in Wireless sensor network using neural network. In *International Conference for Convergence for Technology-2014*, pp. 1-4, (2014, April).

[12] Mekni, T., Taarit, K. I., Ksouri, M. Neural network based congestion detection protocol. In *2017 International Conference on Internet of Things, Embedded Systems and Communications IINTEC*, pp. 201-205 (2017, October). DOI: 10.1109/IINTEC.2017.8325939

[13] Rezaee, A. A., Pasandideh, F. A fuzzy congestion control protocol based on active queue management in wireless sensor networks with medical applications. *Wireless Personal Communications*, vol. *98*(1), pp 815-842 (2018). DOI 10.1007/s11277-017-4896-6

[14] M. Sugeno and G. T. Kang. Structure identification of fuzzy model. Fuzzy Sets and Systems, vol. 28 (1), pp 15-33 (1988). DOI: 10.1016/0165-0114(88)90113-3

[15] Al -Hmouz, A., Shen, J., Al –Hmouz, R. Modeling and simulation of an adaptive neuro-fuzzy inference system (ANFIS) for mobile learning. *IEEE Transactions on Learning Technologies*, vol. 5(3), pp. 226-237 (2011). DOI: 10.1109/TLT.2011.36

[16] Paul, S. K., Azeem, A., Ghosh, A. K. Application of adaptive neuro-fuzzy inference system and artificial neural network in inventory level forecasting. *International Journal of Business Information Systems*, vol. 18 (3), pp. 268-284 (2015). DOI: 10.1504/IJBIS.2015.068164

[17] Jang, J.S.R., Sun, C.T., Mizutani, E. Neuro-Fuzzy and Soft Computing. Prentice Hall (1997).

[18] Walia, N., Singh, H., & Sharma, A. ANFIS: Adaptive neuro-fuzzy inference system-a survey. *International Journal of Computer Applications*, *123*, 13p (2015). DOI: 10.5120/ijca2015905635.

[19] Khairnar, M., Vaishali, D., Kotecha, D. Simulation-based performance evaluation of routing protocols in vehicular ad-hoc network, vol.3 (10) (2013). ISSN 2250-3153

Takoua MEKNI received her engineering degree in industrial and automatic computing from the National Institute of Applied Science and Technology (INSAT) in 2014. And she is currently pursuing her PhD degree in the National Engineering School of Tunis. Her research interests include wireless sensor networks supervision using artificial intelligence.

Kaouther Ibn Taarit is currently an Assistant Professor at the National Engineering School of Monastir (ENIM). She received her engineering degree in Electrical Engineering in 2006, her MS degree in Control and Signal Processing in 2007. In 2010, she obtained her PhD degree from the National Engineering School of Tunis (ENIT) and Central School of Lille (EC-Lille). Her research interests include time delays systems, hybrid systems, parametric identification problems, diagnosis of dynamic systems and Internet of Things (IoT).

Moufida Ksouri is a professor at the National Engineering School of Tunis. She obtained her Engineering Master in 1982 from the Ecole Normale Superieure de l'Enseignement Technique and her PhD degree in 1999 from the University of Sciences and Technologies, Lille, France. Her research interests cover the system control theory, fault detection and isolation, fault tolerant control, hybrid dynamical systems, time delays systems and Internet of Things (IoT).