A Spike Neural Controller for Traffic Load Parameter with Priority-Based Rate in Wireless Multimedia Sensor Networks

Nadia Adnan Shiltagh
Assistant professor Dr.
Computer Engineering/ Baghdad University
E-mail:dr-nadiaat@hotmail.com.

Marwa Taher Naser
Computer Engineering/Baghdad University
E-mail: marwammmmm_2003@yahoo.com.

ABSTRACT

Wireless Multimedia Sensor Networks (WMSNs) are a type of sensor network that contains sensor nodes equipped with cameras, microphones; therefore the WMSNS are able to produce multimedia data such as video and audio streams, still images, and scalar data from the surrounding environment. Most multimedia applications typically produce huge volumes of data, this leads to congestion. To address this challenge, This paper proposes **Modify Spike Neural Network control for Traffic Load Parameter with Exponential Weight of Priority Based Rate Control algorithm (MSNTLP with EWBPRC)**. The Modify Spike Neural Network controller (MSNC) can calculate the appropriate traffic load parameter μ for each parent node and then use in the EWPBRC algorithm to estimate the transmission rate of parent nodes and then assign a suitable transmission rate for each child node. A comparative study between (MSNTLP with EWBPRC) and fuzzy logic controller for traffic load parameter with Exponential Weight of Priority Based Rate Control algorithm (FTLP with EWBPRC) algorithm shows that the (MSNTLP with EWBPRC) is more efficient than (FTLP with EWBPRC) algorithm in terms of packet loss, queue delay and throughput. Another comparative study between (MSNTLP with EWBPRC) and EWBPRC with fixed traffic load parameter (μ) shows that the MSNTLP with EWBPRC is more efficient than EWBPRC with fixed traffic load parameter (μ) in terms of packet loss ratio and queue delay. A simulation process is developed and tested using the network simulator _2 (NS2) in a computer having the following properties: windows 7 (64-bit), core i7, RAM 8GB, hard 1TB.

Keywords: - wireless multimedia sensor network, congestion control, spike neural network, QoS.

المسيطر العصبي المتصاعد لمعامل حمل المرور بالأعتماد على الاولوية لتحديد معدل الأرسال في الشبكات الآسلكية ذات الوسائط المتعددة

أ.م.د. نادية عدنان شلتاغ
هندسة الحاسبات/جامعة بغداد

مروى طاهر ناصر
هندسة الحاسبات/جامعة بغداد

الخلاصة

شبكات الأستشعار اللآسلكية ذات الوسائط المتعددة (WMSNs) هي نوع من شبكات الأستشعار التي تحتوي على عقد أستشعار مجهزة بكاميرات وميكروفونات، ولذلك فإن WMSNs تكون قادرة على إنتاج بيانات الوسائط المتعددة مثل تيارات الفيديو والصوت ، و الصور الثابتة، والبيانات العددية من البيئة المحيطة. ومعظم تطبيقات الوسائط المتعددة عادة ما تنتج كميات ضخمة من البيانات,مما يؤدي إلى الأزدحام في WMSNs. لمواجهة هذا التحديد ,تم أقتراح خوارزمية المسيطر

العصبي المتصاعد المطورة لمعامل حمل المرورمع خوارزمية الوزن الأسي للسيطرة على معدل الأرسال بالأعتماد على الأولوية (MSNTLP with EWBPRC) . ال (MSNC) يمكنة حساب القيم المناسبه لمعامل حمل المرور μ لعقدة الوالد ومن ثم أستخدامها في خوارزمية **EWPBRC** لتخمين معدل الأرسال لعقد الوالد ومن ثم تعيين معدل أرسال مناسب لكل عقدة طفل أستنادا إلى معامل حمل المرور μ وأولوية كل عقدة. وقد أظهرت دراسة المقارنة بين (MSNTLP with EWBPRC مع خوارزمية مسيطر المنطق الضبابي لمعامل حمل المرورمع خوارزمية الوزن الأسي للسيطرة على معدل الأرسال بالأعتماد على الأولوية (FTLP with EWBPRC) أن (MSNTLP with EWBPRC) هو أكثر كفاءة من خوارزمية (FTLP with EWBPRC) في ناحية خسارة الحزم , تأخير طابوروالانتاجية. وتبين مقارنة أخرى بين خوارزمية (MSNTLP with EWBPRC) مع (EWBPRC مع معامل الحمل الثابت (μ)) أن (MSNTLP with EWBPRC) هو أكثر كفاءة من خوارزمية (EWBPRC مع معامل الحمل الثابت (μ)) من ناحية الخساره الحزم ,وتأخير الطابور .تم تطوير عملية المحاكاة والأختبار بأستخدام برنامج NS2 في حاسوب يحمل المواصفات التالية: نظام تشغيل النوافذ7 (64 بت) ,المعالج i7، ذاكرة الوصول العشوائي 8GB، القرص الثابت 1TB.

الكلمات الرئيسية: شبكات الأستشعار ذات الوسائط المتعددة , التحكم بالأزدحام, الشبكة العصبية المتصاعدة, جودة الخدمه

1. INTRODUCTION

Wireless sensor networks (WSNs) have drawn the interest of the research community during the past years. This growing attention can be significantly attributed to new applications enabled by large-scale networks. Small devices are able of collecting information from the physical environment, performing the simple processing of the extracted data, and forwarded it to a remote place. Most deployed wireless sensor networks measure scalar data, such as pressure, temperature, humidity, or the place of objects. Generally, most of the applications, wireless sensor network has low bandwidth requirements and are commonly delayed tolerant. Because available cheap device like cameras and microphone that are capable to capture multimedia content from the surrounding environment has strengthened the growth of Wireless Multimedia Sensor Networks (WMSNs). A Wireless Multimedia Sensor Network (WMSN) is defined as a network of wireless integrate device that able retrieval video, audio streams and still images from the physical environment**.** WMSNs will also be capable to process, store in real-time, and merge multimedia data that produce from different sources **Thomas, et al., and 2010**. There are many different resource constraints in WMSNs involve energy, data rate, buffer size, memory and sensors processing ability due to the physically small size of the sensor and the natures of the multimedia applications which generate large amounts of data, **Si-Yeong Bae, et al., 2014**. This may cause congestion in the sensor nodes. Therefore, wireless multimedia sensor network will not be able to ensure the demands of the quality of service (QoS). For this reason, it is necessary to establish a traffic control mechanism to achieve a high level of throughput, the minimum transmission delay and decrease number of retransmission to increase network lifetime, **Sukhchandan, 2014**. According to the reason of the congestion occurs. There are two types of congestion as shown in **Fig. 1A** and **Fig.1B**. The first type is Node –Level congestion It occurs when the packet service rate is smaller than the packet arrival it causes buffer overflow in the sensor node. This type of congestion can lead to increased queuing delay and packet loss. Packet loss will increase retransmission and will increase consume energy. The second type is Link-Level congestion it occurs when Wireless channels are participated by several nodes using CSMA protocols (Carrier Sense Multiple Access). Collisions may happen when several active sensor nodes try to use the channel simultaneously. Link level congestion leads to decreasing the overall throughput, and waste energy at the sensor nodes because of increase packet service time and decrease link utilization, **Omid, et al., 2012**.

2. NETWORK MODEL

In **Fig.2** a Wireless multimedia Sensor Network consists of four parents (node 1, node 2, node 3 and node 4) sensor nodes, six child sensor node and one sink node. All these are distributed in the area in a tree topology. There are some properties of this model as followed: - All sensor nodes can generate rate randomly. All sensor nodes in the network model don't have the mobility. The sensor node has two jobs; nodes can generate data traffic, as well as route data traffic originated by other nodes. In this network model all sensor nodes can generate various classes of traffic such as real time (**RET**) multimedia stream (audio or video) or alarm data, this type of traffic need to reach a human in real time, and also loss tolerant, i.e. video streams can tolerate ascertain the level of distortion. The real time traffics need to have high throughput and minimum delay time. Therefore should be assigned a high priority for it. And non -real time traffic class can be classified into three class high priority non-real time traffic, medium priority non-real time traffic and low priority non-real time traffic (**NRET1**, **NRET2**, **and NRET3**). In this type of traffic the delay time is not important (delay tolerant) and needed low throughput.

$$TH\_RET \geq TH\_NRET1 \geq TH\_NRET2 \geq TH\_NRET3 \quad (1)$$

$$DE\_RET \leq DE\_NRET1 \leq DE\_NRET2 \leq DE\_NRET3 \quad (2)$$

Where TH represents throughput, DE represents delay time

For example multimedia sensor node 9 can be generated low non- real time traffic (**NRET3**) only, multimedia sensor node5 can be generated two types of traffic (**RET**, **NRET2**) and multimedia sensor node 8 can be generated four types of traffic (**RET**, **NRET1**, **NRET2**, **NRET3**). Each sensor node has a single path to forward data to the sink node. And each sensor node should be send data traffic generated within the limited channel capacity of the network channel.

The number of queues in wireless multimedia sensor network relies on an application demands, but increasing number of traffic type lead to increase the number of demand queues furthermore, the hardware demands should be increased also. In general the priority queue is widely used in diverse application like operating system and real time system, In fact the priority queue provides low delay time for high priority traffic type thus, the packet with high priority leaves queue first with neglect the order of arrival

The queuing model of each node is shown in **Fig.3.** Each wireless multimedia sensor node has a separate queue for each traffic type. And each multimedia sensor node inserts a traffic type identifier for each local sensor packet to distinguish between traffic types. In actuality, the identifier represents as traffic type for each packet. When a parent node received packets from child node, the parent node will be sent these packets to appropriate queues based on identifier type, this operation represent as processing already happened in classifier unit, then the priority queue scheduler has been provisioned to schedule the diverse traffic with different priority from the priority queues.

3. THE PROPOSED APPROCH BASED ON MODIFY SPIK NEURAL NETWORK

The proposed approach used Modified Spike Neural control for Traffic Load Parameter (MSNTLP) in the parent node to obtain optimum value TLP (μ) that use later with the EWBPRC algorithm to estimate the transmission rate of parent nodes and then assign a suitable transmission rate for each child node based on the traffic load parameter and priority of each child node shown in **Fig.4**. The simple structure of the proposed controller in the parent nodes is shown in **Fig.5** Where $\mathrm{R}_{\mathrm{g}}^{\mathrm{n}}(\mathrm{t})$ is rate generating by parent node (n) at time t, $\mathrm{R}_{\mathrm{g}}^{\mathrm{n}}(\mathrm{t}-1)$ is rate generating by parent node (n) at time (t-1) and E (t) is the difference between them. E (t) is calculating be Eq. (3) and μ is the traffic load parameter.

$$E\ (t) = R_g^n(t) - R_g^n(t-1) \tag{3}$$

The congestion control unit in the proposed (MSNTLP with EWBPRC) algorithm shown in **Fig.6** consists of three units:-

When $\mathrm{R}_{\mathrm{g}}^{\mathrm{n}}(\mathrm{t})$ enter to the CDU, the CDU is used to calculate error E (t) and then send this error to MSNTLP to find the suitable TLP (μ) value and then the Rate Adjustment Unit (RAU) calculate a new rate for each node depend on the EWBPRC algorithm, TLP (μ) value from MSNTLP and source priority of each node. At the end the congestion notification unit (CNU) uses implicit notification signal to indicate the new rate for all active nodes in the network.

The structure of the forward spike neural network with modified Spike-Prop training algorithm is shown in **Fig.7** is consisted of three layers, one neuron in the input layer (H), ten neurons in the hidden layer (I) and one in the output layer (J). For each connection between each neuron consist of four synapses as shown in **Fig.8.** The number of neurons in the hidden layer (I) and number of synapses in each sub connection are selected by trial and error.

The feed forward phase in modified spike-pro training algorithm is beginning from encoded processes. The real information E (t) is encoded with respect to time by using the equation of encoded Eq (4), **Yesim, et al, 2013.**

$$t_h^f = T_{max} - round(\,T_{min} + \frac{(REI-RE_{min})(T_{max}-T_{min})}{(RE_{max}-RE_{min})}\,) \tag{4}$$

Where t_h^f is actual firing time for input layer, $REI\ is\ real\ input$, RE_{max} is the maximum number of real information, RE_{min} is the minimum number of real information, T_{max} is the maximum time interval and T_{min} is the minimum time interval (measured in msec).

In general the input layer (H) represent as encoding layer and feed forward operation begins from hidden layer (I) by checking each neuron (i) in hidden layer (I) at sequence, if neuron (i) generate spike or not. If neuron (i) in hidden layer (I) generates a spike within a time interval the algorithm of spike neural check the next neuron (i+1) in the hidden layer (I), but if the neuron (i) of hidden layer (I) doesn't generate a spike within a time interval, the algorithm should calculate the member potential (internal state) $m_i(t)$ for each neuron (h) in the input layer (H) using equation Eq (5) **Yesim , et al, 2013.** by depending on input spikes t_h^f of input layer (I).

$$m_i(t) = \sum_{h=1}^{NH} \sum_{K=1}^{D} w_{hi}^K(R)\, \varepsilon(t - t_h^f - d^K) \tag{5}$$

Where NH is Number of neurons in the input layer, D is the number of sub-synaptic, $w_{hi}^K(R)$ symbolizes to weights of hidden layer connection for certain input R, d^K symbolizes the delay of the connection and $\varepsilon(t - t_h^f - d^K)$ represent as response function. When member potential (internal state) exceeds the threshold value (ϑ), then neuron generates a spike during the time interval (t), else it will be reset in the next time (t+1).

After finishing check all neurons (i) in the hidden layer (I), the same operation of the algorithm of spike neural applies on the neuron (j) in the output layer (J), but the time spike of neuron (i) in hidden layer (I) represent as input to output layer (J), the member potential $m_j(t)$ is calculate be Eq (6) **Yesim, et al, 2013.**

$$m_j(t) = \sum_{i=1}^{NI} \sum_{K=1}^{D} w_{ij}^K(R)\, \varepsilon(t - t_i^f - d^K) \tag{6}$$

Where NI is number of neuron in hidden layer, $w_{ij}^K(R)$ symbolizes to weights of output layer connection for certain input R and t_i^f is actual firing time for hidden layer.

After feed forward processing finished, the backward phase begins where the synaptic weights of connection will be updated. Opposed to feed forward the back propagation begins from output layer and return back to the hidden layer. The synapse weights of output layer will be updated according to Eq. (7), Eq. (8), Eq. (9), **Yesim, et al, 2013.** and Eq. (10).

$$\frac{\partial}{\partial t}(y_i^K) = y_i^K \left(\frac{1}{t - t_i^f - d^K} - \frac{1}{\tau} \right) \tag{7}$$

Where $\partial/\partial t\,(y_i^K) = 0$ for $t \leq t_i^f + d_K$

Where y_i^K is Unweight contribution of single synaptic terminal.

$$\delta_j = \frac{T_{j-}t_j^f}{\sum_{i=1}^{NI} \sum_{K=1}^{D} w_{ij}^K(R) \frac{\partial}{\partial t}(y_i^K)} \tag{8}$$

$$\Delta w_{ij}^K(R) = \eta.\delta_j.y_i^K \tag{9}$$

$$w_{ij}^K(R+1) = w_{ij}^K(R) + \Delta w_{ij}^K(R) \tag{10}$$

Where δ_j is Delta function and η learning rate.

The synapse weights of hidden layer will be updated according to Eq. (11), Eq. (12), **Yesim, et al, 2013.** and Eq. (13).

$$\delta_i = \frac{\sum_{j=1}^{NJ} \delta_j \sum_{K=1}^{D} w_{ij}^K (R) \frac{\partial}{\partial t} (y_i^K)}{\sum_{h=1}^{NH} \sum_{K=1}^{D} w_{hi}^K (R) \frac{\partial}{\partial t} (y_h^K)} \tag{11}$$

$$\Delta w_{hi}^K(R) = \eta . \delta_i . y_i^K \tag{12}$$

$$w_{hi}^K(R+1) = w_{hi}^K(R) + \Delta w_{hi}^K (R) \tag{13}$$

The spike pro training algorithm is repeated until the Root Mean Square Error (RMSE) is less than the desired value of error. The learning rate η will be updated after the end of each epoch according to the Eq. (14) **Huijuan, et al. 2012.**

$$\eta(epoch+1) = \begin{cases} A . \eta(epoch) \ if \ \ error(epoch+1) < error(epoch) \\ B . \eta(epoch) \quad if \ error(epoch+1) > C . error(epoch) \\ \eta(epoch) \quad if \ error(epoch+1) = error(epoch) \end{cases} \tag{14}$$

Where A, B and C are constant values.

Then coding processing is starting to convert the output encoding information to real information (RE (t_j^f)) by deriving from Eq (4).

$$RE(t_j^f) = \frac{\left(T_{max} - t_j^f - T_{min}\right) \times (RE_{max} - RE_{min})}{(T_{max} - T_{min})} + RE_{min} \tag{15}$$

4. RATE MANAGEMENT ALGORITHM

The transmission rate adjustment in the propose approach consists of three steps:-

Step1: computing the new output transmission rate of sink node

The EWPBRC is used to estimate output transmission rate of sink node for the next round $\mathrm{R}_{sink_{out}}(\mathrm{t}+1)$, where $\mathrm{R}_{sink_{out}}(\mathrm{t})$ is represented as the output transmission rate of sink node at time (t). $\mathrm{R}_{\mathrm{sink(in)}}(\mathrm{t})$ is represented as the total input rates of sink node at time (t), which can be calculated by summation of all output transmission rate of the child nodes.

$$R_{sink_{out}}(t+1) = R_{sink_{in}}(t) . (1\_\lambda) + \lambda \ . \ \mathrm{R}_{sink_{out}}(\mathrm{t})\ ) \tag{16}$$

Where λ is a constant value within the range $0 \leq \lambda \leq 1$.

Step 2: allocate new transmission rate for the child node

In WMSN will be supplied with various types of multimedia sensor node, and a multimedia sensor node will be distributed at associated with geographical location according to the various levels of importance.

In the proposed approach each multimedia sensor node has two types of priority class: traffic class priority ($\mathrm{P}_{\mathrm{TR}}^{\mathrm{n}}$) and geographical location priority($\mathrm{P}_{\mathrm{G}}^{\mathrm{n}}$), which can be installed manually by the network manager.

Let $\mathrm{P}_{\mathrm{TS(s)}}^{\mathrm{n}}$ represent as traffic source priority which can be installed manually according to service differentiation, P_{TR}^{n} can be calculate by the summation of all individual traffic source priority in the source node (n), when a node (n) has highest traffic source priority therefore it will be a high traffic class priority. And can be calculated by Eq (17), Young**, et al. 2014.**

$$P_{TR=}^{n} \sum P_{TS(s)}^{n} \quad (17)$$

Where s represents a traffic source class in the source node (n), therefore s should be [RET, NRET1, NRET2 and NRET3].

According to the propose approach, transmission data based on geographical location and type of traffic class therefor it will be put appropriate priority to geographical location, to calculate total priority TP^{n} of multimedia sensor node (n) follow as Eq (18) Young**, et al. 2014.**

$$TP^{n} = p_{G}^{n} \ . \ P_{TR}^{n} \quad (18)$$

GOP^{n} Represent as global priority of multimedia sensor node (n), can be calculated by following Eq (19), Young**, et al. 2014.**

$$GOP^{n} = \sum_{k \in c\ (n)} GOP^{k} + TP^{n} \quad (19)$$

Where c (n) is a group of child nodes of a node (n).

In fact the value of global priority is different between child sensor node and parent sensor node because the global priority in child node equal to total priority only because the child node don't have any child node, for example node 6 don't have any child node, the global priority of it is equal to total priority only, but node 2 has two child nodes (node 5 and node 6) therefor, the global priority can be calculated by Eq (19).

To ensure that the algorithm will participate the current capacity only among active nodes therefor the global priority is calculated only for active multimedia sensor nodes that is meant the value of total priority is equal to zero if traffic source is not active. For example, when node 2 need to calculate global priority according Eq (19) and the same time node 6 don't generate any traffic source therefor the global priority of node 6 is equal to zero, for this the global priority of node 2 can be calculated by summation of the total priority of node 2 and global priority of node 5 at assume it is generating traffic source at this time.

To calculate global priority for sink node $\mathrm{GOP}^{\mathrm{Sink}}$, should be summation all global properties of all child nodes $\mathrm{GOP}^{\mathrm{k}}$ of sink node, and is represented as follows:

$$GOP^{sink} = \sum_{k \in C(sink)} GOP^{k} \tag{20}$$

By using the proportion of the global priority of child nodes GOP^{n} and the global priority of sink node $\mathrm{GOP}^{\mathrm{Sink}}$ to distribute the output transmission rate of sink node $\mathrm{R}_{\mathrm{Sink(out)}}$ between all child nodes, to calculate new output transmission rate for node n $\mathrm{R}_{\mathrm{n(out)}}$.

$$\mathrm{R}_{\mathrm{n(Out)}} = \mathrm{R}_{\mathrm{Sink}_{\mathrm{Out}}} \frac{\mathrm{GOP}^{\mathrm{n}}}{\mathrm{GOP}^{\mathrm{Sink}}} \tag{21}$$

Step3: adjust the new output transmission rate of each parent node by using MSNTLP

In the propose approach, the parent node is responsible of distributed output transmission rate between all child nodes by adjusting the traffic load parameter (μ) in each parent node by using MSNC, as shown in **Fig.8.** In the proposed approach each child node have one parent node that means there is a single path from source to destination.

The input transmission rate for each parent node $\mathrm{R}_{\mathrm{n(in)}}$ Comes from summation all output transmission rates of the related child nodes $R_{k(out)}$, follow as Eq (22)**.**

$$R_{n(in)} = \sum_{\mathrm{k \in c\,(n)}} \mathrm{R}_{\mathrm{k(out)}} \tag{22}$$

The output transmission rate for each parent node comes from the summation of the input transmission rate of parent node $\mathrm{R}_{\mathrm{n(in)}}$ and the transmission rate generating by parent node.

$$R_{n(out)} = R_g^n + R_{n(in)} \tag{23}$$

To find the different transmission rate in parent node is can be calculated by Eq (24), Young**, et al. 2014.**

$$\Delta R_{(n)} = \mu . R_{n(out)} - R_{n(in)} \tag{24}$$

Where μ is the traffic load parameter (TLP) will be adjusted by MSNC. Because when the μ is fixed, a large change in data transmission causes a significant difference between the input transmission rate and the estimated output transmission rate of each sensor node.

The output transmission rate of child nodes, as follow as Eq. (25) .Young**, et al. 2014.**

$$R_{k\,(out)} = R_{k\,(out)} + \Delta R_{(n)} . \frac{GOP^{k}}{GOP^{n}} \tag{25}$$

5. SIMULATION RESULTS

The proposed algorithm is implemented using the Network simulator 2 (NS-2)**.** To test the effectiveness of the proposed approach and the performance of the network. The QoS parameters: throughput, Packet Loss Ratio and queue delay. The simulation model consists of ten multimedia sensor nodes and one sink node, as shown in **Fig.9**. The transmission route of that model is a single-path transmission (static routing). The transmission data collected by multimedia sensor nodes are generated randomly; all senders transmit data traffic to a single sink node. In the simulation model, there are four types of traffic class, namely RET, NRET1, NRET2 and NRET3. The simulated traffic is **Pareto**. Each packet size is 500 bytes, the buffer size of each child node is set up as 50 packets, and the buffer size of the sink node is 100 packets. All multimedia sensor nodes start to send their data at the start of the simulation rounds and stop at the end of simulation rounds. The simulation rounds are 100 seconds. The channel capacity of each multimedia sensor node is set to the same value: 2 Mbps. The combination of traffic classes that each node could have in the simulation model is shown in Table (1), where PW represent as the Priority Weight of traffic, assuming that all sensor nodes have the same geographical priority equal to 1. Suppose the assigned priority for RET, NERT1, NERT2, and NERT3 classes are equal to 4, 3, 2 and 1, respectively. For example, the sensor node 8 which has all traffic classes, the traffic class priority, is 10 (4 + 3 + 2 + 1 = 10), while for sensor node 5 which has NRET1and NRET2 traffic classes, the traffic class priority, is (4+3=7).

The performance of the proposed algorithm is compared with FLCTLP with EWBPRC and EWPBRC with fixed traffic load parameter algorithms μ=0. 9. The performance is evaluated mainly according to the following metrics:

1-Throughput: measured in terms of number of packets received during simulation time.

2-Packet Loss ratio: it is a percent between the number of packets lost in the network and the number of packets generated by the sensing nodes.

3-Average Queue delay: it is the ratio between the time difference between the incoming and outgoing packet rates in the queue to total number of received packets.

5.1 Throughput

The total input transmission rate for node1 comes from the total output transmission rate of all his child nodes. **Fig.11** shows the simulation results of the input transmission rate for node1 in each round by using FTLP with EWBPRC algorithm.

Fig.12 shows the input transmission rate of node1 that comes from the total output transmission rate of all his child nodes by using MSNTLP with EWBPRC algorithm.

Fig.13 shows a comparison of the input transmission rate of node1 for the FTLP with EWPBRC algorithm and the MSNTLP with EWPBRC algorithm. Simulation result shows that the FTLP with EWPBRC algorithm has an input transmission rate smaller than that of the MSNTLP with EWPBRC algorithm.

Fig.14 shows the input transmission rate of node2 that comes from the total output transmission rate of all his child nodes by using FTLP with EWBPRC algorithm.

Fig.15 shows the input transmission rate of node2 that comes from the total output transmission rate of all his child nodes by using MSNTLP with EWBPRC algorithm.

Fig.16 shows a comparison of the input transmission rate of node2 for the FTLP with EWPBRC algorithm and the MSNTLP with EWPBRC algorithm. Simulation result shows that the FTLP with EWPBRC algorithm has an input transmission rate smaller than that of the MSNTLP with EWPBRC algorithm.

As it is evident from Fig.13 and Fig.16 the proposed approach performs better than FTLP with EWBPRC algorithms to increase the input transmission rate of node1 and node2. Because the proposed approach use modifies spike neural to regular the traffic load parameter (μ) therefor this can reduce differences in transmission rate between the estimated output transmission rate and input transmission rate show in Eq (24).

5.2 Packet Loss Ratio

In WMSN application having a low packet loss is very important. When data is lost, the sender will need to retransmit the lost data. Thus, leading to consume the remaining energy, and reduction in the total lifetime of the network.

Fig.17 shows a comparison of the packet loss ratio for three algorithms, namely, the WPBRC algorithm with μ =0. 9, the FTLP with EWPBRC algorithm and MSNTLP with EWRBPRC. From the simulation result, it can be seen that the proposed algorithm has the lowest loss. From **Fig.17**, one can observe that the proposed approach has an average packet loss ratio better than the average packet loss ratio of EWBPRC with fixed value μ (traffic load parameter) because the fixed value of TLP (see in Eq 24) causes a large variation in data transfer rate between input transfer rate and the estimated output transfer rate of each sensor node. Where the proposed approach can regulate TLP (μ) by using MSNC to decrease sending rate of the sensor node when the traffic load increase in the network or increase the sending rate of the sensor node When the traffic load low in the network. One can also notice that FTLP with EWBPRC perform well in some rounds to minimizing the packet loss ratio, but it is not better than MSNTLP with EWBPRC because the proposed algorithm use MSN with adaptive learning rate to find μ value and this give us optimum value (μ).

5.3 Average Queue Delay

Delay is defined as the duration since a packet enters a queue of a beginning node until it arrives at the ending node. The definition of beginning and ending nodes depends on the type of delay. Queue Delay is the time the packet need to wait before begin transmit because the queue was not empty when it arrived to queue.

Fig.18 shows a comparison for the average queue delay of the EWPBRC algorithm with μ = 0.9, the FTLP with EWPBRC algorithm, and the MSNTLP with EWPBRC algorithm. Simulation result shows that the NEWPBRC algorithm has an average queue delay smaller than that of the EWPBRC algorithm with μ=0. 9 and the (FTLP with EWPBRC) algorithm. The proposed algorithm use MSNC to find optimum values (μ), where the (EWBPRC with a fixed value of μ (0.9)) cannot effectively distribute transmission rate among sensor nodes when the traffic congestion increase. One can also observe that FTLP with EWBPRC perform well in some rounds to minimizing the queue delay, but it is not better than MSNTLP with EWBPRC because the proposed algorithm use SNN with adaptive learning rate to find μ value and this give us optimum value (μ).

3. CONCLUSION

In this paper, MSNTLP with EWPBRC algorithm is implemented in WMSN using the NS2 simulator. Then take average of performance parameters to get a more accurate result. The proposed algorithm enhances the transmission rate and tries to avoid congestion. The proposed algorithm provides good QoS in terms of minimizing queue delay, packet loss ratio and increasing throughput. The Average queue delay achieved by the proposed algorithm is lower by (4.98%) than that achieved by FTLP with EWPBRC algorithm. The proposed algorithm can be (11.8%) less average packet loss than FTLP with EWPBRC algorithm. The proposed algorithm can be (8.97%), (4.45%) large average throughput than FTLP with EWPBRC algorithm in node1and node2 respectively. And because use Modify Spike Neural Network based on modified spike-pro training algorithm lead to speed up the learning rate of SNN and minimized the error between actual and desired value so the training of SNN does not fall in the local minimum value.

REFRENCES

- Abbas. A. R, Mohammad. H. Y and Amir. M. R, 2014, *Optimized Congestion agreement Protocol for Healthcare Wireless Sensor Networks*, Wireless Personal Communications An International Journal, Spring, ISSN: 0929-6212, Vol 75, No 1, P.P 11-34
- Huijuan. F, Jiliang, L., and Fei, W, July 2012, *Fast Learning in Spiking Neural Networks by Learning Rate Adaptation*, Chinese Journal of Chemical Engineering, P.P 1219-1224.
- Si. Y. B, Kyoung. W. P, Sung. K. L and Jin.G. K, 2014,*Traffic Control Mechanism Considering Packet Priority in Wireless Multimedia Sensor Network*, Contemporary Engineering Sciences, Vol.7, No. 24, P.P. 1363 – 1371.
- Sukhchandan. R, 2014, *Research Challenges in Wireless Sensor Network: A State of the Play*, Northwest Institute of Engineering & Technology, India.
- Thomas. D, Qing. P, Gordon. W.S and Natarajan. M, 2010, *On the Design and Implementation of Wireless Multimedia Sensor Networks*, International Journal of Next-Generation Networks, USA, Vol.2, No.3, PP 20-26.
- Yesim. O, Okyay. K and Rahib. A, March 2013, *Spiking Neural Networks for the Control of a Servo System*, IEEE International Conference on Mechatronics (ICM), Vicenza, P.P 94-98, 1 March 2013.
- Young. L. C and Hung. P. L, 2014, “A fuzzy logical controller for traffic load parameter with priority-based rate in wireless multimedia sensor networks ”, Applied Soft Computing, Elsevier, P.P. 594–602 .

List of symbols

d^K= delay of specific synaptic, msec.

E (t)= error.

$\mathrm{GOP}^{\mathrm{sink}}$ = global priorities of the sink node.

$\mathrm{GOP}^{\mathrm{n}}$ = the global priority.

$\boldsymbol{h}$ = neuron sequence in input layer.

$\boldsymbol{i}$ = neuron sequence in hidden layer.

$\boldsymbol{m_i(t)}$ = membrane potential of hidden layer neurons.

$\boldsymbol{NH}$ = number of neurons in the input layer.

$\mathbf{P_G^n}$ = geographical priority.

$\mathbf{P_{TR}^n}$ = traffic class priority.

$\mathbf{P_{TS(s)}^n}$ = the traffic source priority S in sensor node n.

$\boldsymbol{REI}$ = real input pattern.

$\mathbf{R_g^n(t)}$ = rate generating by parent node, Kbit/sec.

$\mathbf{R_{k\,(out)}}$ = the output rate of the kth child of parent node n, Kbit/sec.

$\mathbf{R_{n(out)}}$ = the output transmission rate of node n, kbit/sec.

$\mathbf{R_{n(in)}}$ = the input transmission rate of node n, kbit/sec.

$\mathbf{R}_{\boldsymbol{sink_{out}}}$ = the output transmission rate of sink node, kbit/sec.

$\mathbf{R}_{\boldsymbol{sink_{in}}}$ = the input transmission rate to sink node, kbit/sec.

$\boldsymbol{t}$ = current time, msec.

$\boldsymbol{t_h^f}$ = actual spiking time of neurons at input layer, msec.

$\mathbf{TP^n}$ = the total priorities of node n.

$\boldsymbol{w_{hi}^K}$ = weight between input and hidden layer.

$\boldsymbol{y_i^K}$ = unweight contribution of single synaptic terminal.

$\boldsymbol{\mu}$= traffic load parameter.
$\boldsymbol{\eta}$ = learning rate.

$\boldsymbol{\vartheta}$ = threshold value for spiking.

$\boldsymbol{\delta_j}$ = delta function.

Table.1 Simulation parameters.

Total area	500 meter X 500 meter
Network Model	• Ten Sensor Nodes. • One Sink Node.
Transmission Routing	Single Path Transmission. • Multi-Hop. • Many-to- One.
Traffic Class	• Real time • High priority non real time • Medium priority non real time • Low priority non real time
Traffic Type	Pareto
Packet Size	500 Bytes
Queue Size	• Childe Node 50 Packets. • Sink Node 100 Packets.
Channel Bandwidth	2 Mbps
Simulation time	100s
Round	30

Table 2. The state of traffic classes in each sensor node.

Sensor Node NO.	RET (PW=4)	NRET1 (PW=3)	NRET2 (PW=2)	NRET3 (PW=1)	P_{TR}^{n}
Node 1	ON	ON	OFF	OFF	7
Node 2	OFF	ON	OFF	OFF	3
Node 3	ON	OFF	ON	OFF	6
Node 4	OFF	ON	ON	OFF	5
Node 5	ON	ON	OFF	OFF	7
Node 6	OFF	ON	ON	OFF	5
Node 7	OFF	ON	OFF	ON	4
Node 8	ON	ON	ON	ON	10
Node 9	OFF	OFF	OFF	ON	1
Node 10	ON	OFF	OFF	ON	5

Table 3. SNN specification.

Parameter	**Value**	**Unit**	**Description**
Network topology	1,10,1	Unit less	Number of units in Input/Hidden/output
$\boldsymbol{\eta}$	0.003	Unit less	Learning rate value initially
$\boldsymbol{A, B, C}$	1.10 , 1.04 , 0.8	Unit less	Constants of adaptive learning rate
$\boldsymbol{\tau}$	6	msec	Time constant
$\boldsymbol{\vartheta}$	1.5	Unit less	Threshold value for spiking
$\boldsymbol{D}$	4	Unit less	Number of delay synapses per connection
$\Delta\boldsymbol{t}$	0.5	msec	Step size
$\boldsymbol{w}$	[0-1]	Unit less	Initial weights of hidden and output layer

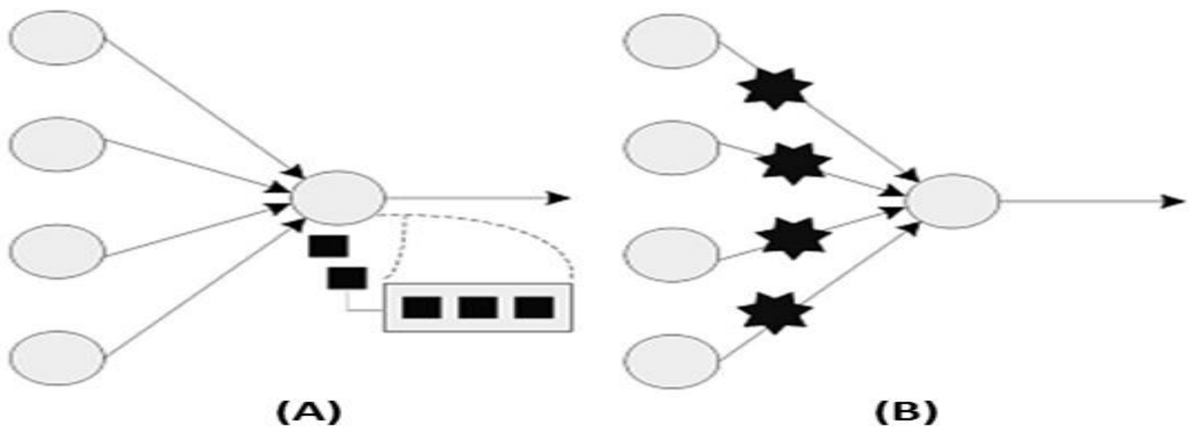

Figure 1. Congestion type **Omid, et al., 2012**.

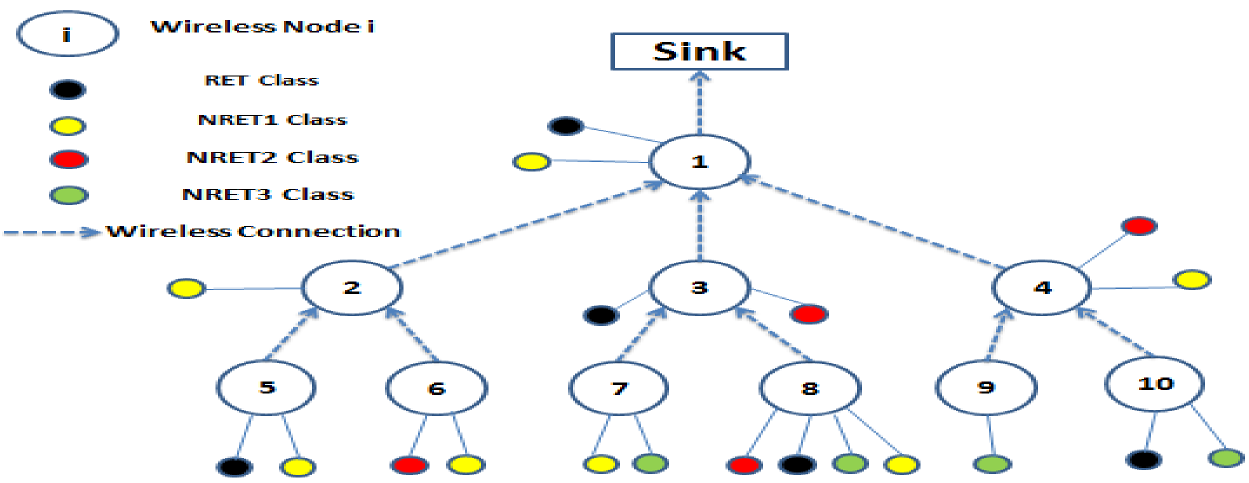

Figure 2. WMSN model **Young, et al. 2014.**

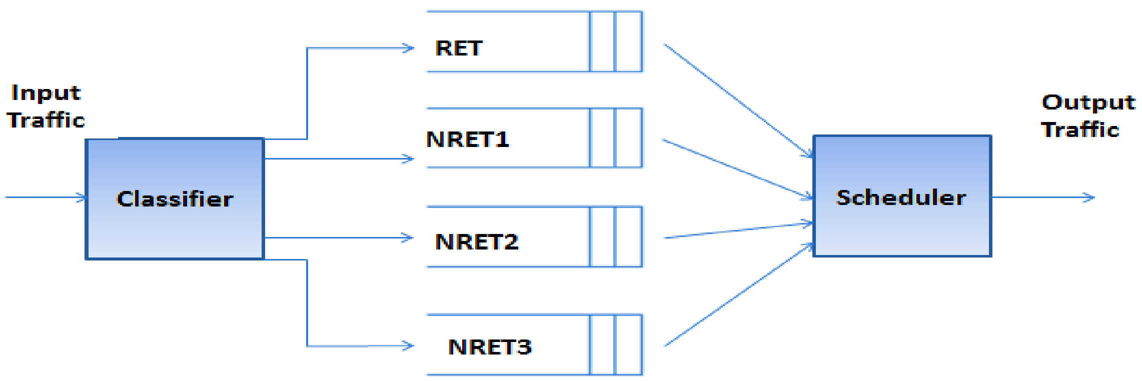

Figure 3. Queue model for each sensor node**.**

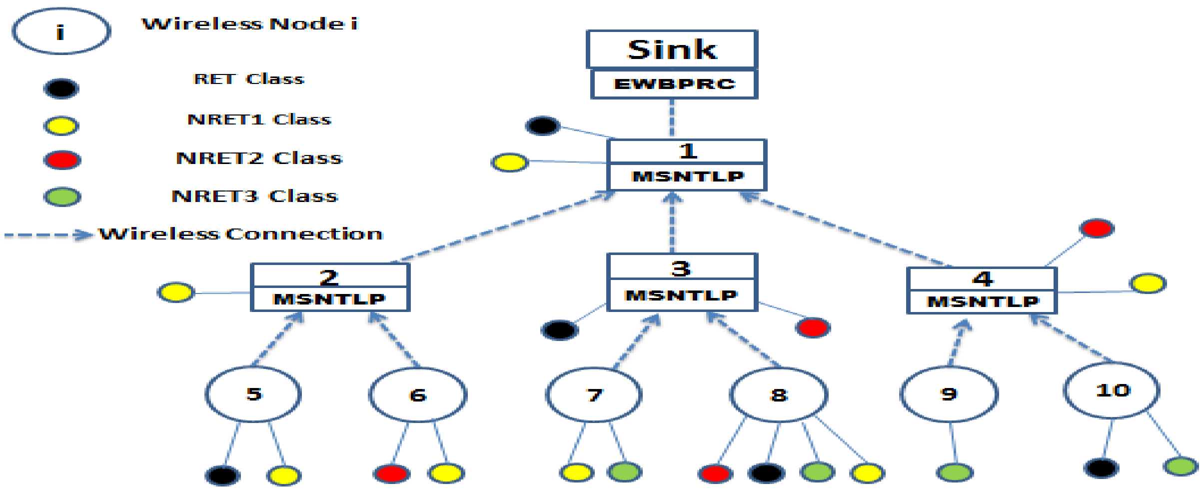

Figure.4 MSNTLP traffic class model in WMSN **Young, et al. 2014.**

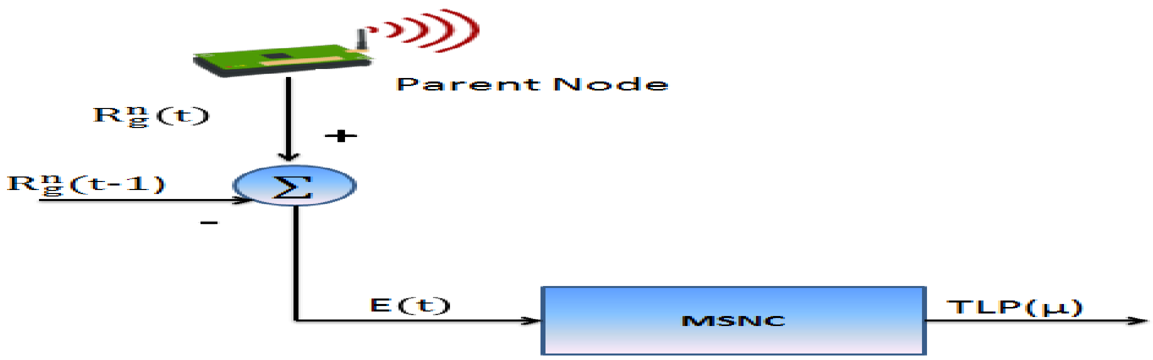

Figure 5. Block diagram of MSNTLP.

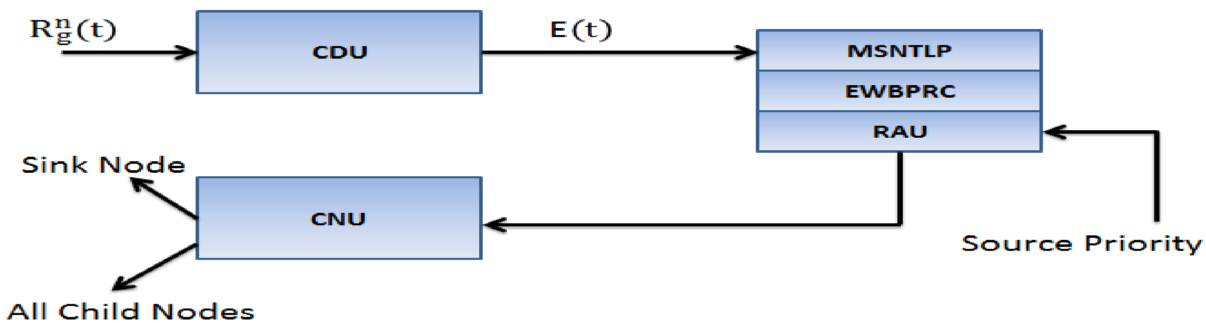

Figure 6. Structure of MSNTLP with EWPBRC congestion control unit.

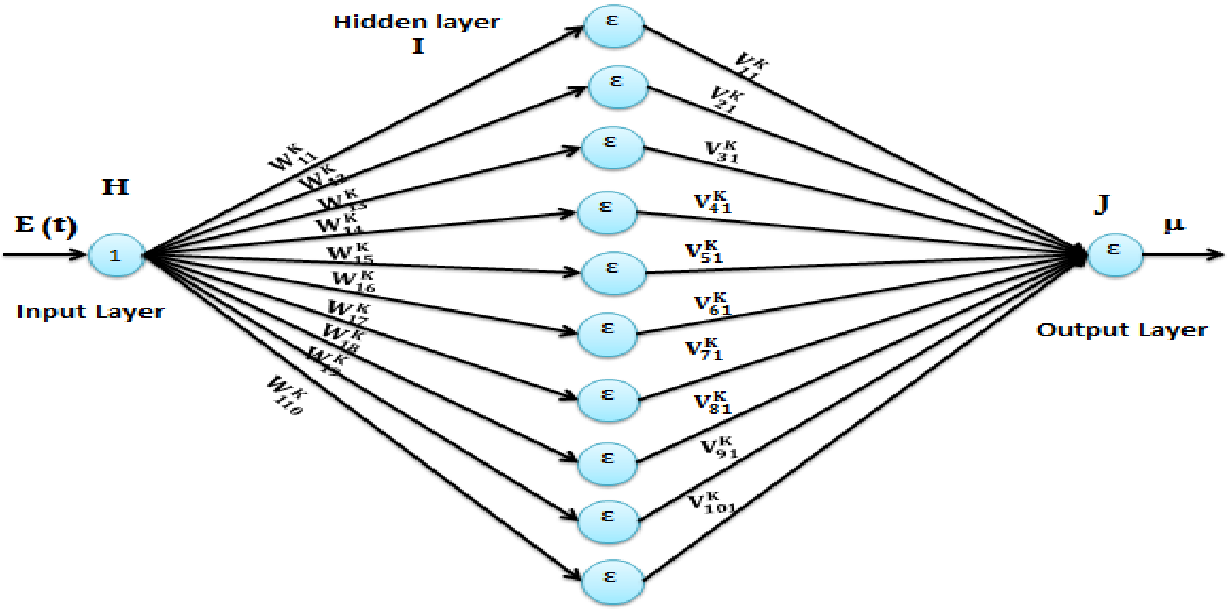

Figure 7. Structure of SNN.

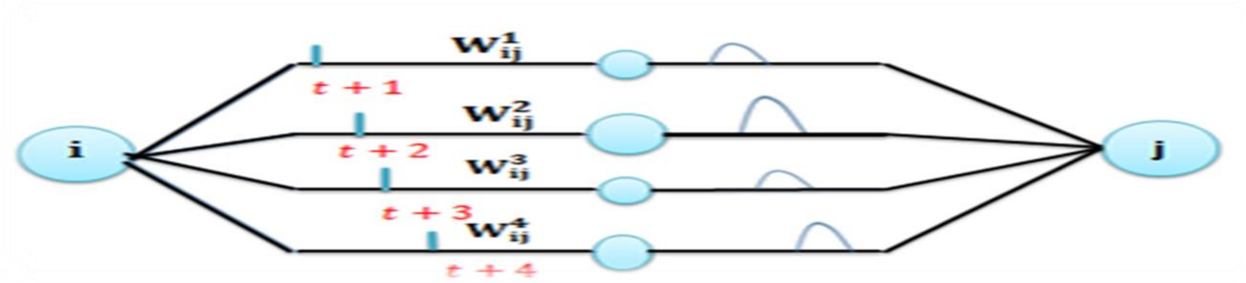

Figure 8. Sub-connection consist of four synapses.

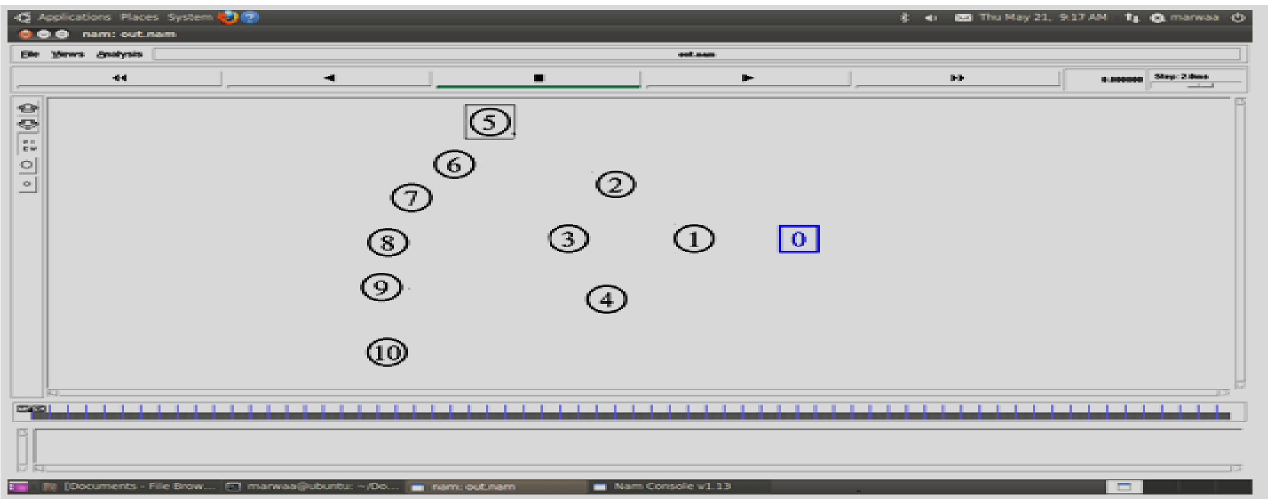

Figure 9. Simulation model using NS2.

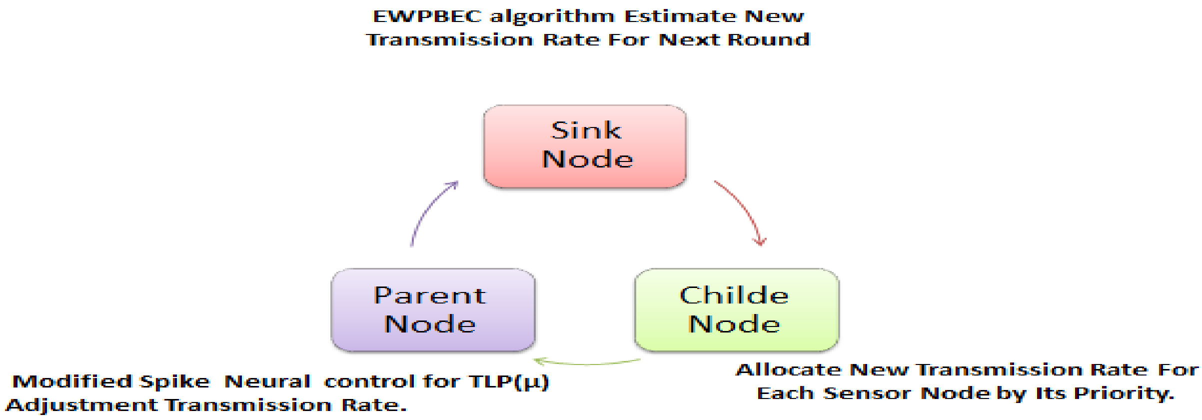

Figure10. Rate management for the (MSNTLP with EWBPRC) algorithm.

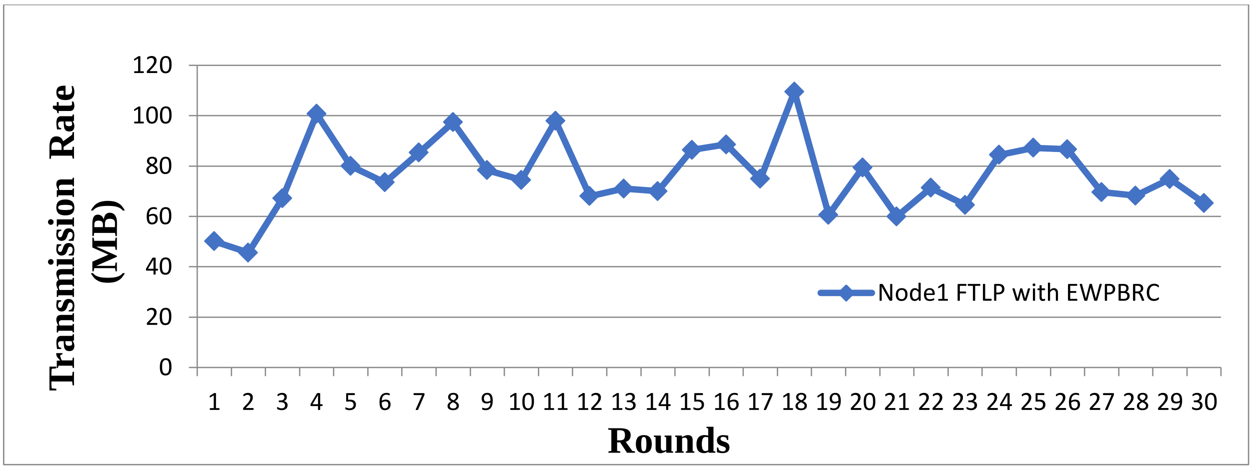

Figure11. Input transmission rate of node1 over rounds by using the (FTLP with EWPBRC) algorithm.

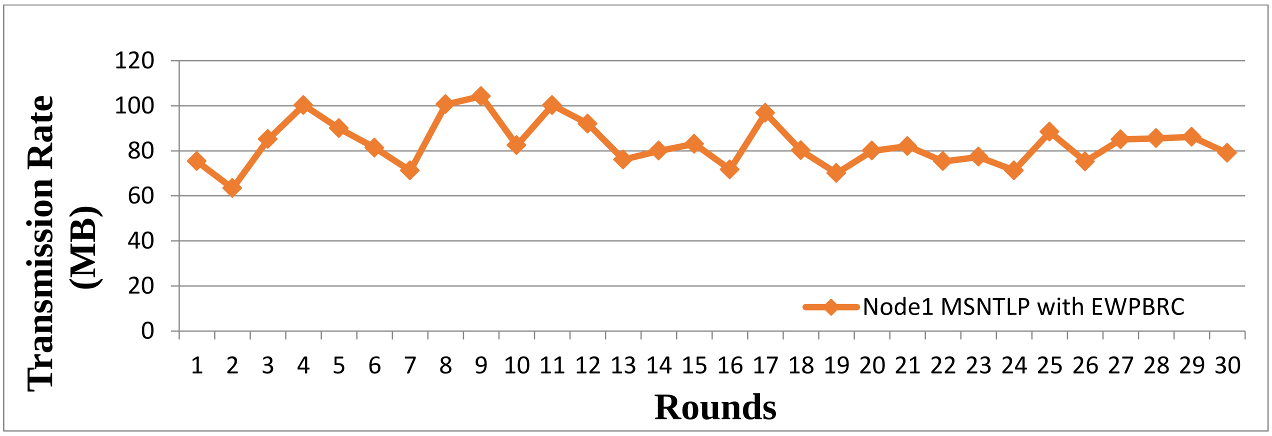

Figure12. The input transmission rate of node1 over rounds by using the MSNTLP with EWPBRC algorithm.

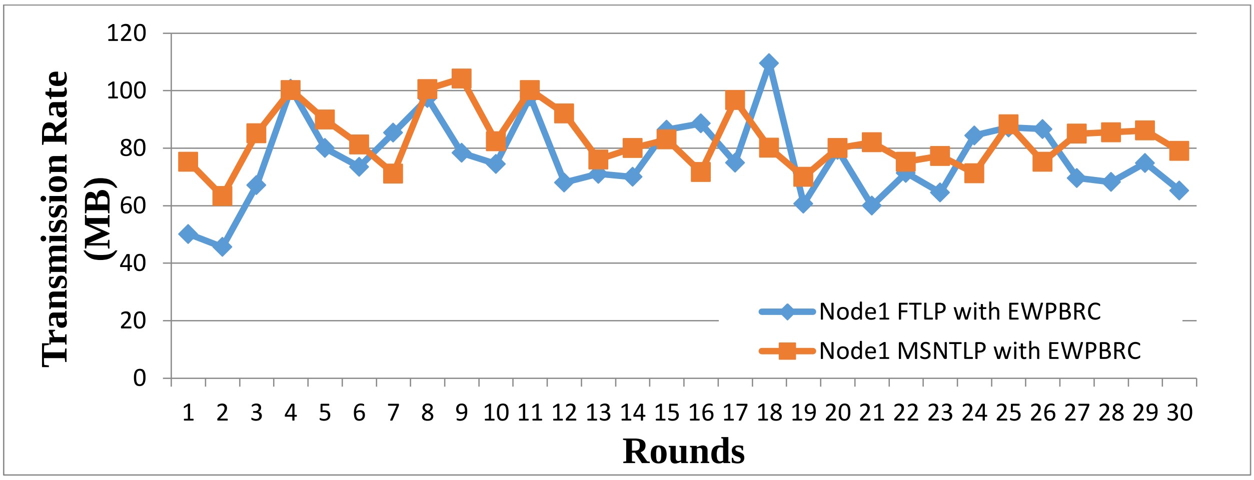

Figure13. Comparisons of input transmition rate of node1for different algorithms.

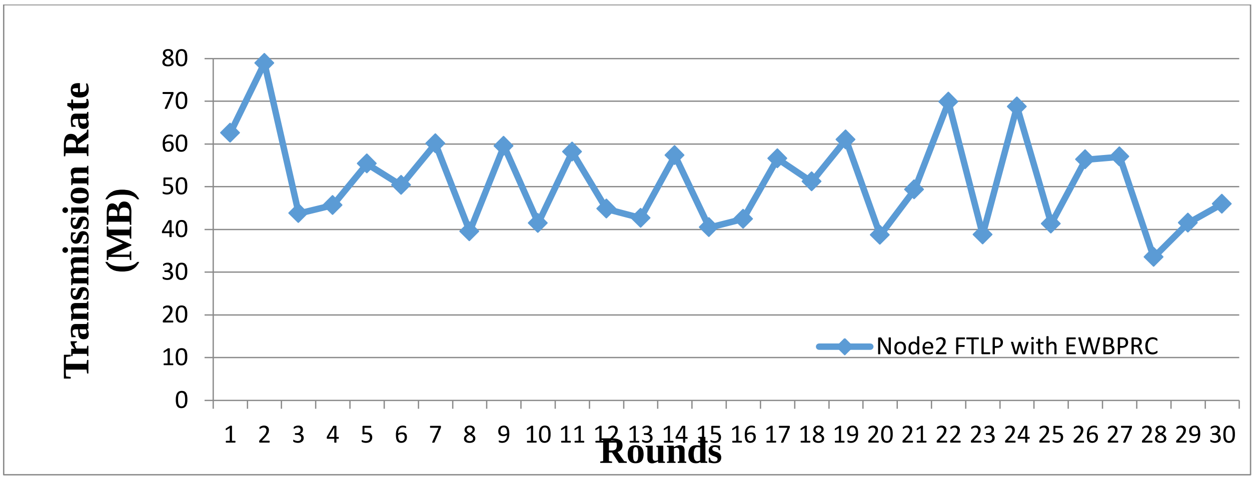

Figure14. The input transmission rate of node2 over rounds by using the (FTLP with EWPBRC) algorithm.

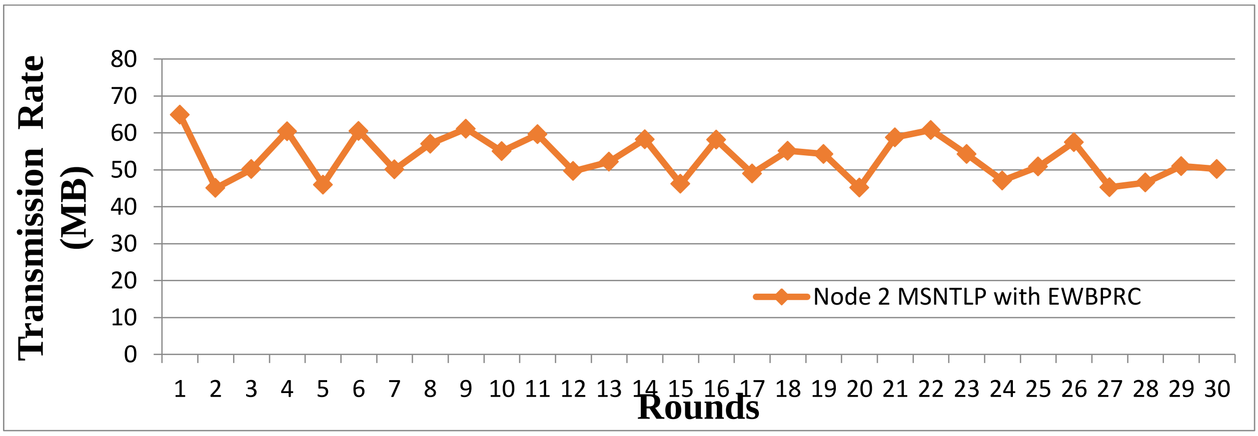

Figure15. Input transmission rate of node2 over rounds by using the (MSNTLP with EWPBRC) algorithm.

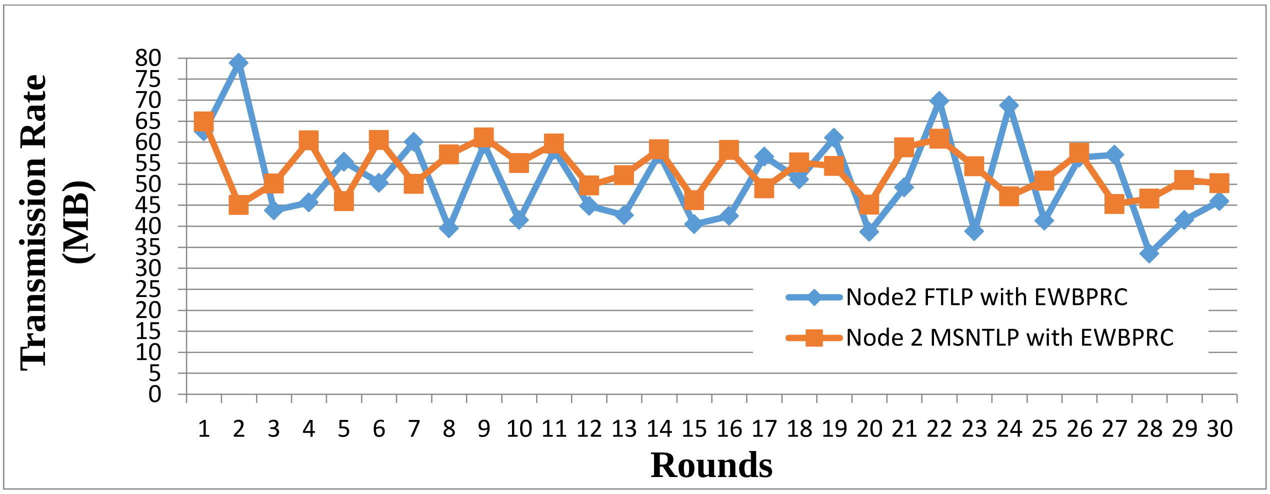

Figure16. Comparisons of input throughput of node2 for different algorithms.

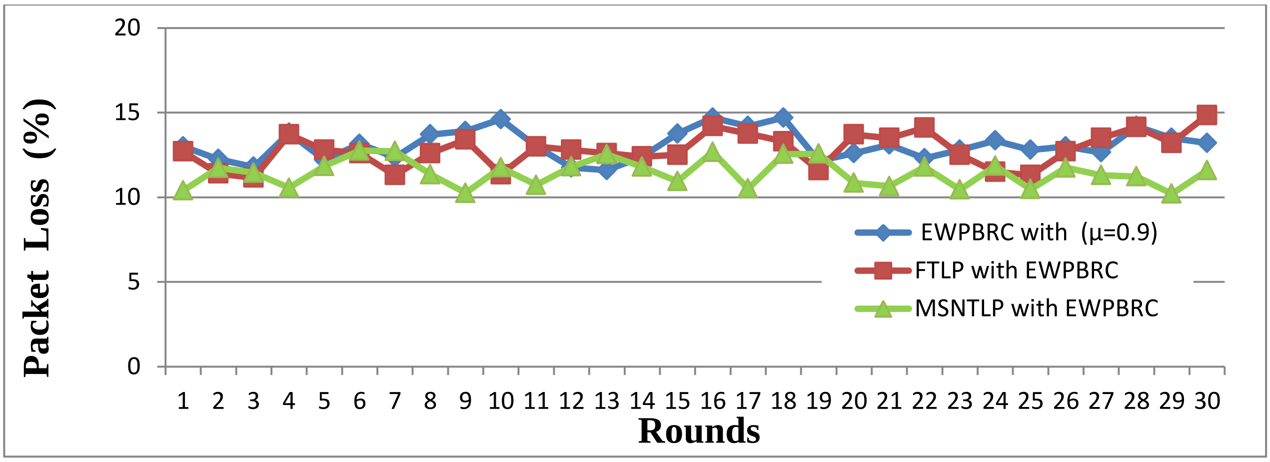

Figure17. Comparisons of packet Loss Ratio for Different Algorithms.

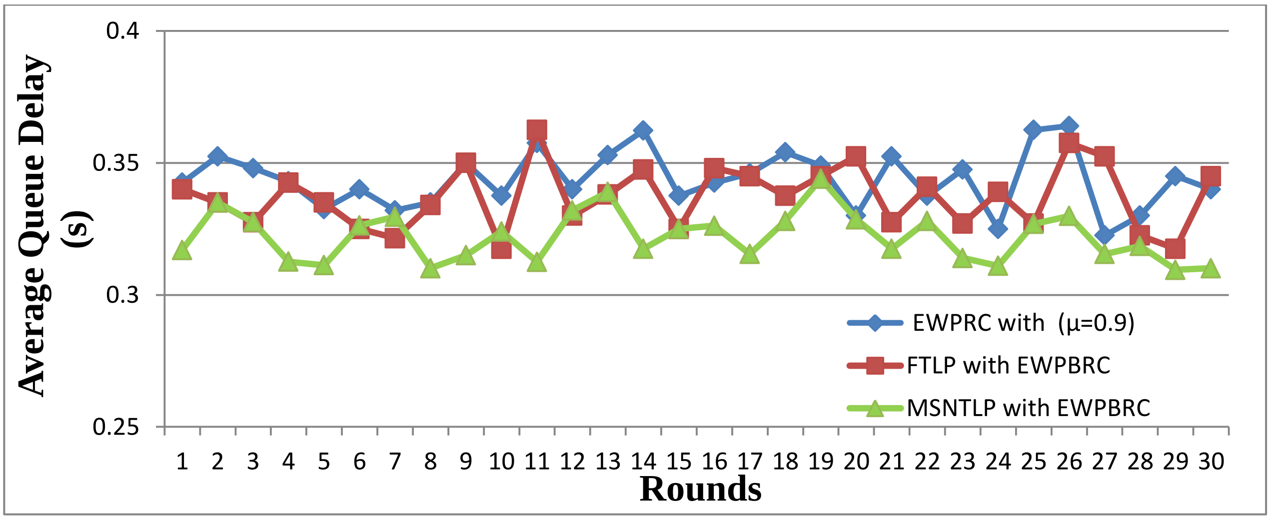

Figure 18. Compared the Average Queue Delay for Different Algorithms.