A Fuzzy Based Approach for Rate Control in Wireless Multimedia Sensor Networks

Mohammad Hossein Yaghmaee Moghaddam[i *] and Hamid Reza Hassanzadeh[ii]

Received 10 February 2012; received in revised 12 March 2012; accepted 01 May 2012

ABSTRACT

Wireless Multimedia Sensor Networks (WMSNs) undergo congestion when a link (or a node) becomes overpopulated in terms of incoming packets. In WMSNs this happens especially in upstream nodes where all incoming packets meet and directed to the sink node. Congestion in networks, if not handled properly, might lead to congestion collapse which deteriorates the quality of service (QoS). Therefore, in order to avoid such situations corresponding actions should be taken into account so that to yield lower packet loss and consequently energy loss that is of utmost importance in WSNs. However, the term "packet loss" as implied by today's literature might not be effective in many applications especially in multimedia sensor networks. In this paper a new weighted packet loss metric is proposed which is best suited for multimedia sensor networks that convey packets of different priority classes. The proposed method then tries to minimize the aforementioned criterion by means of fuzzy queue management and a newly introduced adaptive rate control mechanism, in the presence of both abrupt and gradual changes in network dynamics. The employment of these two techniques provides us a synergy to handling short term and long term variations arising through the underlying simulated networks. The simulation results approve the superiority of the proposed approach over the selected competitive method when dealing with packets of different priorities.

KEYWORDS

Adaptive rate control, Multimedia sensor networks, Fuzzy queue management

1. INTRODUCTION

Wireless Sensor Networks (WSNs) [1] may include a vast number of sensor nodes which are densely deployed to carry sensing data pertaining to environmental conditions. These networks share outstanding characteristics and concerns among which resource constraints and power consumption are of the most critical ones that if cared delicately improve the quality of service (QoS) significantly.

The field of wireless Multi Media Sensor Networks (WMSNs) [2], on the other hand, is considered as a specific case of WSNs in which the data to be delivered towards the sink node is generated from multimedia sensors such as cameras and microphones. Due to the high transmission rates of wireless multimedia sensors, a great part of the energy resources is spent for disseminating large number of packets that make WMSNs vulnerable towards congestion. The congestion, if not mitigated in the right way, might beget subsequent packet losses and deteriorates the quality of service which in part wastes energy resources and shortens the network lifespan. The packet loss might also prolong the packet transfer time which is also of immense importance for multimedia data. Based on the aforementioned reasons, developing efficient protocols and mechanisms to maximize network lifetime while improving the quality of service at the same time has been remained a demanding task in wireless multimedia sensor networks. Generally, in applications of WMSNs, a sensor node consists of different sensors [2] that generate data with different levels of importance. Due to these characteristics of wireless multimedia sensor networks the incoming data should be serviced keeping these priorities in mind. Therefore, taking advantage of the so-called differentiated service architecture in the foregoing networks is considered an indispensable task.

The other behavior that WMSNs share is that some nodes may produce bursty traffic when confronting to an unpredictable situation. This might be as a result of

[i] [*] Corresponding Author, M.H. Yaghmaee Moghaddam is with the Department of Computer Engineering and Center of Excellence on Soft Computing and Intelligent Information Processing, Ferdowsi University of Mashhad, Mashhad, Iran (e-mail: yaghmaee@ieee.org).

[ii] Hamid Reza Hassanzadeh is a graduate student in Department of Computer Engineering, Ferdowsi University of Mashhad (FUM), Mashhad, Iran (e-mail:ha.hassanzadeh@ieee.org)

sudden environmental changes in nature where the nodes planted or due to occurrence of an important event in a specific region. In this case most of the traditional QoS methods lack handling the situation properly. A sudden increase in traffic rate through the network might cause queue overflow and congestion in the links which usually come up with higher delays and loss probabilities as a consequence. There is also the possibility that this traffic overflow stays steadily for a while. In that case the maximum transmission rates corresponding to the nodes of the network should be tuned such that the burden of higher traffic rates at a specific area becomes partly alleviated while keeping the resource allocation fairness at a desired level.

In network literature a node or link is called congested when it receives data with rates above its maximum transmission rate. To tackle with the congestion, diverse mechanisms have been proposed yet each of which falls into one of the two categories of congestion control or congestion avoidance [3]. The former tries to manage the network when congestion occurs but have nothing to do with the network management as long there is no congestion detected. The latter, on the other hand, tries to anticipate when a network might experience a congestion and performs required actions in advance. Meanwhile, there are methods which consider both congestion avoidance and congestion control approaches. Active Queue Management (AQM) is one of the most promising categories of algorithms proposed for congestion avoidance. It marks or drops incoming packets based on a probability that is dependent on the state of the underlying queue to prevent buffer overflow. Random Early Detection (RED) [4] as one of the most well-known methods of this kind, drops the receiving packets based on a probability that is directly proportional to the associated queue length. Although it has been widely deployed, it has its own drawbacks [5] too, including problems with parameter tuning (that is, highly contingent upon network status) and oscillation in lengths of queues, to name a few. Note that there also exist other AQM algorithms such as GREEN and BLUE which differ mostly in the applied congestion indicator and the way they address the parameter tuning problem.

Fuzzy Logic theory was put forward by Zadeh [6] in his seminal paper on Fuzzy Sets in 1965. Since his first paper on the subject, the notion of fuzzy sets has been extended in diverse areas such as reasoning, approximation and control. One of the most outstanding strengths of fuzzy logic is that it somehow resembles natural languages in that it can model the imprecise knowledge of human into a mathematical framework which is then applied to different applications such as fuzzy controllers in industry.

As shown in Fig. 1, a fuzzy controller is a system based on fuzzy rules whereby instead of binary logic it rests upon fuzzy logic. These systems are comprised of four main components, as follows:

- **Fuzzifier**
  As fuzzy systems manipulate fuzzy sets, the crisp inputs should be converted into fuzzy ones in some way. In so doing, the fuzzifier takes in a crisp input and returns a fuzzy set depending on the type of the selected fuzzifier.
- **Rule Base**
  All the decisions in a fuzzy system are based on the existing fuzzy rules. These rules are made by combinations of fuzzy antecedents and consequents which are fuzzy sets. The rule base is usually evolved during the time via heuristic approaches or it may be formed by the help of an expert operator who knows the underlying system well.
- **Inference Engine**
  After the crisp input converted into a fuzzy set, it is fed into the inference engine for calculation of the output fuzzy set. The inference engine makes use of the rules embedded into the rule base to come into a fuzzy conclusion. Depending on the choice of inference engine different outputs might be produced.
- **Defuzzifier**
  Finally, to produce outputs that comply with plants' inputs, the resulting fuzzy set should be converted back into a crisp value via a defuzzifier.

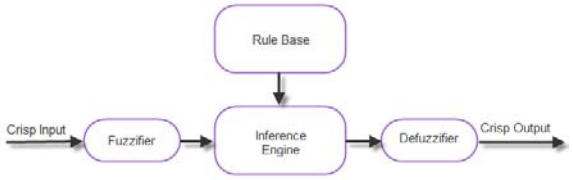

Fig. 1. A Fuzzy System.

The rest of this paper is organized as follows. In section 2, the related works or those that are partially related to ours are described briefly. Section 3 then presents the proposed model. In section 4, we use experimental results to evaluate our proposed model and finally section 5 concludes by taking a glance at the proposed algorithm and introducing new open areas for further research activities.

2. RELATED WORK

In this section, we consider recent articles that are related in some sense to our introduced model. These articles basically fall into two main categories, namely, those that have something in common with fuzzy queue management and those that employ rate adjustment to boost congestion control and consequently quality of service in wireless sensor networks.

In [7] the authors proposed a modified TCP equation-based active queue management mechanism which is based on GREEN algorithm using a fuzzy approach. In [8] a fuzzy controller for tuning incoming packet rates is proposed and named Fuzzy-Multiple Queue Management. It takes in queue lengths associated with priority queues as inputs and generates the maximum allowable input and output traffic rates. In [9] an adaptive fuzzy congestion control solution for active queue management is proposed. It benefits an adaptive mechanism for generation of fuzzy membership functions. The output of the fuzzy controller is the packet rejection probability to prevent buffer overflow, which is achieved through a hierarchical multilevel decision making. The simulation results are then compared with those of fixed membership functions. This method does not consider prioritized packets at all and only considers general TCP networks. In [10] authors propose a hybrid method for congestion detection in TCP networks based on a fuzzy controller and optimize it using a Particle Swarm Optimization (PSO) technique. Like the previous one, this method fails when dealing with multi-media packets.

To date, various congestion control methods have been proposed for wireless sensor networks [11 -14]. Congestion control and fairness (CCF) is proposed in [12] as a distributed and scalable algorithm that eliminates congestion within a sensor network and guarantees the fair delivery of packets to a sink node. According to CCF, each node measures the average rate at which packets can be sent from, shares that rate among the child nodes and recursively continues to the downstream nodes. CCF uses packet service time to deduce the available service rate and congestion information is implicitly reported. It controls congestion in a hop-by-hop manner and each node uses exact rate adjustment based on its available service rate and number of child nods. CCF is known to guarantee simple fairness. As shown in [11], CCF suffers from a non-work-conserving shortcoming in the sense that a parent node should wait for a child until a specific number of packets arrive, after which it switches to another child for another specific amount of packets to be served. In other words, if during a period one of the children does not send any packet, the parent node blocks until the expected number of packets received. This makes the proposed method inefficacious in such scenarios and leads to low throughputs. In [11] a priority based congestion control protocol (the so-called PCCP) is proposed which takes advantage of a new congestion indicator that is defined as the ratio of packet inter-arrival time to the packet service time. Based on the aforementioned metric as well as the node priority index, it introduces a congestion control technique via a hop by hop approach. It has also been shown that PCCP achieves efficient congestion control and flexible weighted fairness for both single-path and multipath routing. Although PCCP has proven to be a promising method, further analysis has been disclosed the fact that for the case of low congestion, the proposed method increases the scheduling rate as well as the source rate for all traffic sources regardless of their priority. Besides, the PCCP considers the mere geographical priority and cannot make a distinction between traffics of different classes. In [15] a new approach motivated by the PCCP and CCF is proposed which addresses their existing weaknesses. It considers different priorities for each node based on both the importance level of the incoming traffic along with the node's geographical priority. It also adjusts sending rates of the traffic sources based on the nodes' congestion status and the introduced priority index. This resolves the PCCP problem for low congestion situations. The method serves as the selected approach for later comparisons in our work.

3. PROPOSED MODEL

The approach cited in [15] although outperforms considerably in respect of the two predecessors (namely, PCCP and CCF), has its own drawbacks. Firstly, it does not take into account the transient characteristics of multimedia traffics in the sense that an abrupt increase in an incoming traffic's rate is always assumed to yield a congestion while this might not be true in many cases. Consider a scenario whereby an important event triggers (this could be a critical alarm) in a region where a sensor node has been planted that gives rise to generation of a bursty traffic for a short period of time. Due to the queue management policy adopted by the authors, a great number of the incoming packets will be discarded due to traffic overflow. The problem with the mentioned approach is not limited to short term traffic rate variations rather, undesired situations may also arise when the changes in incoming rates are permanent or gradually increased/decreased. Imagine a scenario where due to a change in environmental conditions, the local traffic sources of some specific sensor node increase their packet generation rates drastically and stay steady in the new configuration during some period of time. While the associated increase in other existing local traffic sources located inside farther nodes are not comparable to these highly excited ones they are given almost the same share of the remaining bandwidth. As a result, while most of the sensor nodes remain inactive and their allotted bandwidths are wasted, there are some few nodes that are in urgent need for the unused available link capacity through the network. Last but not the least, in that model the traffic sources are forced to follow a packet generation scheme based on their priority indexes. That is, the sensor nodes are presumed to generate traffic rates corresponding to their maximum allowable rates which is

a counter intuitive assumption as the local traffic sources are not under control of the network protocols but instead they are more dependent upon geographical location and characteristics of the sensor traffic.

Put it together, in this paper the proposed approach is targeted at the mentioned shortages of the method introduced in [15] and seeks compensating them in some way. By doing so, our model consists of two major units, namely, Congestion Control Unit and Fuzzy Queue Management Unit. Our strategy is to control the nodes' maximum transmission rates based on the weighted packet loss (defined shortly) to resolve permanent changes in packet generation rates of the local traffic sources while leaving the short term changes up to the queue management part. As in [15], for each node there are as many queues as the number of priority classes assumed. The incoming traffic to a node is first directed into a traffic classifier whose duty is to separate the packets of each class and submit them to fuzzy queue managers which are then either rejected or accepted. Without loss of generality, in this paper we assumed four different priority classes which are as follows: real-time class (RTC), high priority class (HPC), mid-priority class (MPC) and low priority class (LPC). The RTC packets are those that are subject to hard delay constraints and should be delivered as soon as possible. The HPC packets are those with high priorities but they are not as much critical as the RTC packets. In the same way, the MPC packets are those with lower degree of importance in regard of HPC packets but more important than the LPC ones which are located at the bottom of this hierarchy. Note also that in this paper we assume that the network possesses a heterogeneous infrastructure meaning that the local traffic sources inside the nodes are separated and that each local traffic source generates packets of an attributed importance degree.

3.1. Congestion Control Unit

Fig. 2 depicts the deployed architecture for the congestion control unit. As easily inferred, it consists of three main building blocks, namely: Congestion Detection Unit (CDU), Rate Adjustment Unit (RAU) and Congestion Notification Unit (CNU).

The CDU is responsible for detection of congestion in advance based on difference in node's incoming traffic rate and its maximum allowable transmission rate. This might be either a positive or negative value and serves as a congestion index. Given the congestion index, the rate adjustment unit (RAU) then calculates the amount of unoccupied bandwidth and shares it among its local traffic sources and the downstream nodes by means of the congestion notification unit proportional to their corresponding average weighted packet loss, defined bellow:

$$WPL^i(t) = \sum_j PP_j \times Loss_j^i(t) \quad (1)$$

$$AWL^i(t+1) = (1-\alpha) \times AWL^i(t) + \alpha \times WPL^i(t) \quad (2)$$

where the trailing superscripts and subscripts denote the node number and packet priority, respectively, α is a learning rate between 0, 1 and PP_j is the priority of packets belonging to the *j*'th class. Finally, the $Loss_j^i(t)$ is the number of packets belonging to the *j'*th priority class which have been dropped at node *i*, in time period between t and $t-\Delta t$ where Δt is a constant that is defined before the simulation starts. Intuitively, the above mentioned relations state that the more the importance of a packet be, the more cost we incur by dropping it.

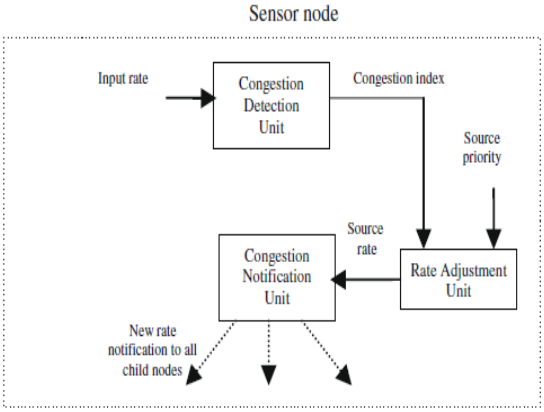

Fig. 2. Congestion control unit.

Fig. 3 shows the pseudo code pertaining to the proposed model. As illustrated in line 2 of the pseudo code, all the links in the network are initialized at the start of simulation. As $Loss_j^i(t)$ is unknown in the beginning, each transmission rate pertaining to a leaf node (i.e., a node which has no child) in the network is initialized based on the number of local traffic sources and their corresponding degrees of importance (the priority of traffic sources). The link rates corresponding to the intermediary nodes are also initialized in the same way but with the key difference that both the node and its downstream nodes are counted recursively. The algorithm continues execution inside the loop until the simulation ends.

```
1.  Proposed_Model{
2.      Initialize_max_transmission_rate ();
3.      last_time=0
4.      While (! Simulation ended){
5.          For (i=1 : numOfNodes){
6.              If (new_packet_arrived(i)=true)
7.                  State=Add_fuzzy_AQM(packet)
8.                  If (State=DROP)
9.                      Re fresh_Loss_j^i(time)
10.         }
11.         If (time-last_time > Δt ) {
12.           Refresh_max_transmission_rate()
13.         }
14.     last_time=time
15.     time=next_time()
16.     }
17. }
```

Fig. 3. Pseudo code for the proposed method.

In each iteration of the loop, new incoming packets are either added to corresponding queues or dropped via the proposed fuzzy AQM in case of packet rejection, the $Loss_j^i(t)$ is updated (lines 7-9).

Finally, the nodes' maximum transmission rates are refreshed after each Δt seconds based on their global priority, GP^i, defined as follows:

$$P^i(t) = AWL^i(t) \times P_{GE}^i \quad (3)$$

$$GP^i(t) = \sum_{k \in C(i)} GP^k(t) + P^i(t) \quad (4)$$

where P_{GE}^i, $GP^k(t)$ and $C(i)$ are the geographical priority, global priority and the set of ith node's children, respectively. In case of having no child, the global priority equals $P^i(t)$. The procedure is done in a bottom-up approach. As easily implied by (3,4), among the nodes which initially have the same global priority those that lose packets with higher priorities during the course of simulation become more important.

3.2. Fuzzy Queue Management

In this section, we propose a new fuzzy active queue management mechanism that is best suited for the case of sensor networks having the presumed architecture. As mentioned earlier, this mechanism is merely applied for managing HPC, MPC and LPC queues, the RPC ones are precluded, however, since they are usually set aside for urgent communications and generate light traffics. Hence, their incoming traffics are served with highest priority via the scheduler meaning that as long as there is a packet waiting in an RTC queue, none of the other queues gets served. Fig. 4 depicts the architecture of the proposed fuzzy queue management. As easily seen from the figure, upon arrival of a new packet, it is fed into a traffic classifier for further determination of its associated priority class via the help of the tag badged to it by the sensor that generated it for the first time. Traffic classifier is the basis of the service differentiation process both in [15] and in the proposed model. Note that, the newly arrived packet might be generated by node's local traffic sources or delivered by a child node. After discriminating the input traffic, the incoming rate associated with each traffic class is measured by the help of rate estimation modules. The computed rates along with the current lengths of all the three queues are taken as inputs by the fuzzy controller. Based on the inputs, the fuzzy controller then decides whether to accept the arrived packet or to reject it. After all, the scheduler module serves the waiting packets in each queue according to its level of priority (i.e. $PP_j \;\; j = 0,...,3$.)

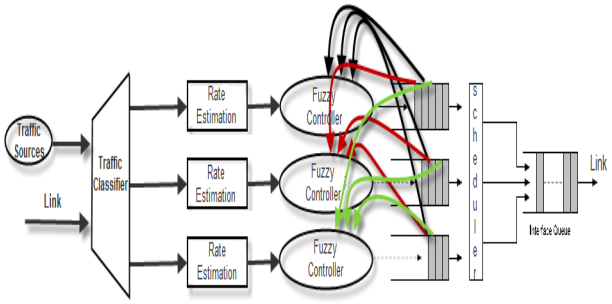

Fig. 4. The architecture of fuzzy queue management.

Table1 demonstrates the decision rules of the employed fuzzy controllers. As directly inferred from the table, when all non real-time queues are very low, low or medium in length and the incoming rate is either medium, low or very low then rejection of any newly arrived packet is improbable (the first row of the table). But this is not the case all the time; consider the last row of the decision table, for example. If the first queue is almost empty but the second is completely full and the incoming traffic rate for the highest priority queue (HPC queue) is high or very high, then regardless of the third queue status, these last two queues should be evacuated as soon as possible so that the scheduler can spend more time on the HPC queue which contains packets of higher priority. This will be demonstrated in experimental results later. Therefore, as opposed to other fuzzy AQM approaches that are prevalent in jargon, here the fuzzy controllers take into account the priority index as well. This is also in harmony with the weighted metrics defined earlier.

TABLE 1. DECISION RULES.

Row #	L1	L2	L3	Incoming Rate of L_i	P1	P2	P3
1	VL/L/M	VL/L/M	VL/L/M	VL/L/M	VL	VL	VL
2	H	VL/L	VL/L	M	M	VL	VL
3	H	VL/L	VL/L	VL/L	VL	VL	VL
4	H	VL/L	VL/L	M	VL	M	H
5	VL/L/M	H	VL/L/M	M	VL	L	M
6	VL/L/M	VH	VL/L/M	L/M	VL	M	M
7	VH	VL/L/M /H/VH	VL/L/M /H/VH	H/VH	VH	VH	VH
8	H	H/VH	VL/L/M /H/VH	H/VH	M	VH	VH
9	H	VL/L/M	VL/L/M H/VH	H/VH	L	L	L
10	VH	VL/L/M /H/VH	VL/L/M /H/VH	M	H	VH	VH
11	VL/L	M	VL/L/M	VH	VL	VH	VH
12	VH	VL/L/M /H/VH	VL/L/M /H/VH	VL/L	M	VH	VH
13	VL/L	VH	VL/L/M /H/VH	H/VH	VL	VH	VH

Finally, Fig. 5 depicts the membership functions corresponding to fuzzy linguistic variables.

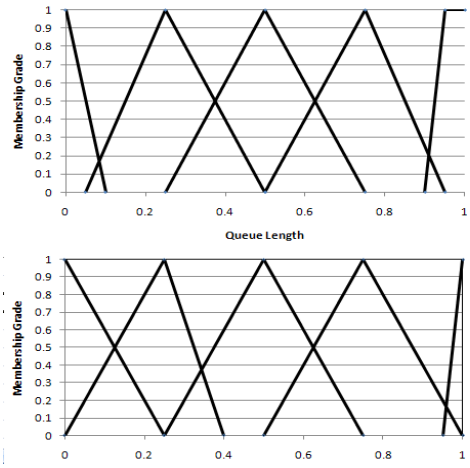

Fig. 5. Fuzzy membership functions for queue lengths (top) and traffic rate (down).

4. EXPERIMENTAL RESULTS

In this section we use computer simulation to evaluate the performance of the proposed model under different scenarios. For this purpose, we simulated a wireless network topology as given in Fig. 6 with 8 sensor nodes. Table 2 illustrates the packet priorities ($PP_j \quad j = 0,...,3$) and Table. 3 demonstrates the local traffic sources available in each node. The results are compared with those of the approach introduced in [15]. In that paper the authors compare the performance of their method in relation to other well-known techniques and in this work we show the superb outperformance of new method over that. To make the results of the methods comparable with each other, the previous method is implemented such that sensors' traffic generation rates are not affected during the rate control process, as already discussed. Clearly from Fig. 6, the selected topology is a rooted tree meaning that there is no multiple paths between two neighboring nodes. The selected fuzzy controller incorporates Mamdani inference engine, as a well-known fuzzy system, with singleton fuzzifiers and center of gravity defuzzifier. To evaluate the performance of both the congestion control unit and the fuzzy queue management unit, the trials are divided into two categories each addressing performance of a specific unit.

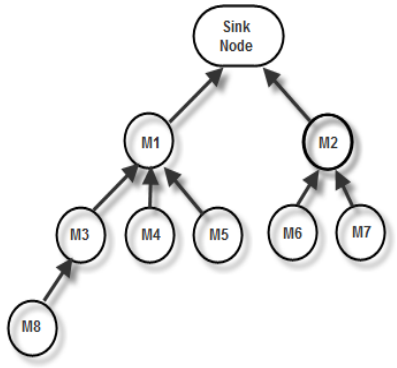

Fig. 6. Network topology of the experimental scenarios.

TABLE 2. DECISION RULES.

	j			
	0	1	2	3
PP_j	10	8	4	1

TABLE 3. LOCAL TRAFFIC SOURCES.

	RTC	HPC	MPC	LPC
M1	1	0	1	0
M2	0	0	1	0
M3	1	1	1	1
M4	1	0	0	1
M5	0	1	1	0
M6	1	0	1	0
M7	0	1	1	0
M8	0	0	1	0

4.1. Evaluation of queue management unit

In this part we evaluate the performance of the proposed fuzzy controller. In the subsequent devised scenario each of the local traffic sources of M7 (see Fig. 6) is forced to produce short term bursts at specific moments. The bursts related to the HPC and MPC traffic sources emerge at t=1.2 sec and t=2.5 sec, respectively, for 2 seconds (as illustrated in Fig. 7). To simulate a more realistic circumstance, the packet generation rate paradigm is arranged to lie in the range of 165 to 175 packets per second. The geographical node priorities have been also set to 1 for all nodes.

Fig. 8 depicts the corresponding plots of M7 queues' lengths during the course of simulation. Notice how in the proposed method the lower priority queue remains empty whenever there are ample of packets waiting for service in the higher priority queues. This is what we intuitively expect for a good queue management algorithm that is to pay more attention to packets of higher level of importance as opposed to previous method where the MPC queue starts augmenting incoming packets as soon as they arrive regardless of the current status of other queues.

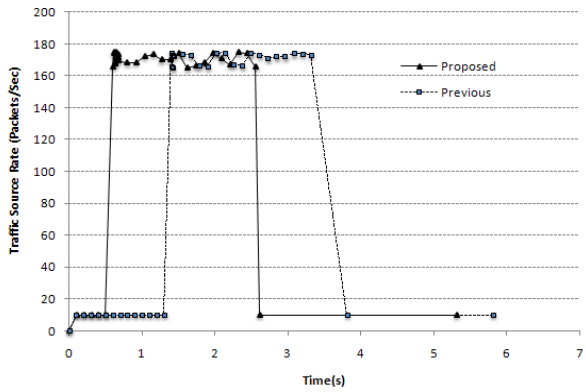

Fig. 7. M7 traffic rates.

Fig. 9 depicts the plot of loss probability and weighted loss probability (a metric which is defined in a similar approach given in (2)) in the aforementioned node. From the figure, it is obvious that the fuzzified algorithm not only has less weighted loss probability but also generally has lower probability to drop the packets during most of the simulation. The plot of total loss probability also follows similar curves. When examining the top and bottom figures closely, it makes evident that that loss probability in the new method exceeds that of the previous slightly, when the bursts happen. This is due to rejection of some of the lower priority packets. However if we consider the figure above, it seems that in terms of weighted loss probability this deterioration in performance partially disappears when considering situation based on a more realistic prioritized perspective.

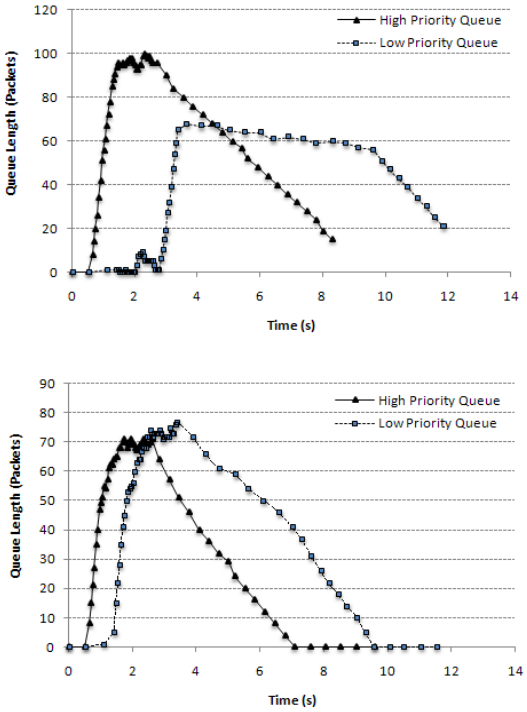

Fig. 8. Queue Lengths corresponding to M7 node, proposed (top), previous (down).

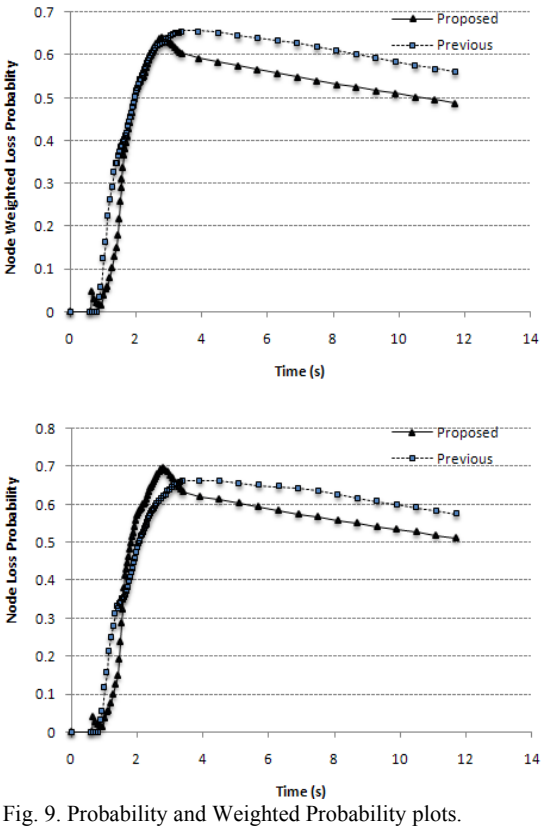

Fig. 9. Probability and Weighted Probability plots.

Fig. 10 shows the overall throughput as well as the individual throughputs associated to each traffic class. According to this figure, the overall throughput for the non-fuzzy method is a bit higher, which as illustrated shortly will be resolved via the rate adjustment policy. However, the weighted throughput is obviously better in the fuzzified method for higher priority classes. The last two subfigures depict the throughput for the HPC and MPC traffics, respectively. Notice how the throughput corresponding to HPC traffic has been improved considerably in the new method but at cost of sacrificing the less important traffics. As alluded earlier, this is also one of the main objectives of the fuzzy controller which seems to be accomplished as desired.

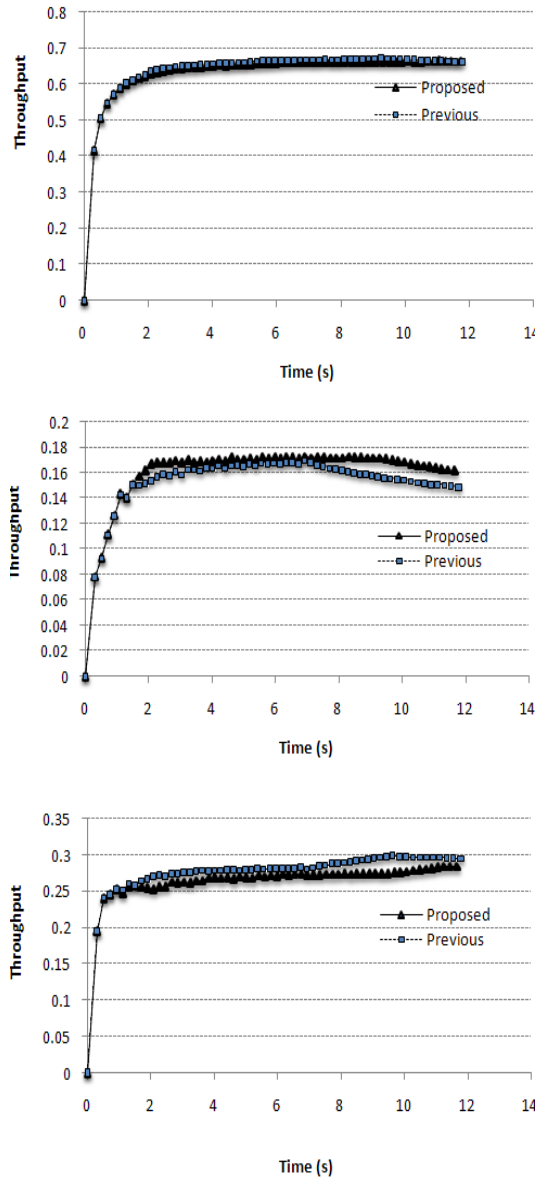

Fig. 10. (from top to bottom) Total, HPC and MPC class throughput.

Finally, Fig. 11 demonstrates the average weighted packet loss during the course of time. From the figure it is implied that fuzzy method outperforms greatly in terms of packet loss during most of the simulation period.

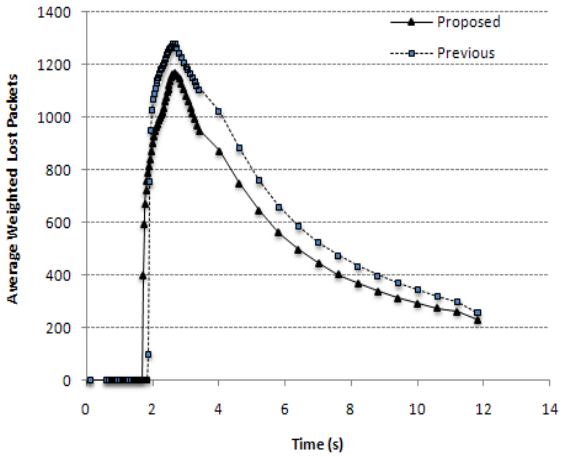

Fig. 11. Average weighted lost packets.

As a final comparison, Table 4 illustrates the number of lost and generated packets distinguished by the four traffic classes.

TABLE 4. OVERVIEW OF LOST/GENERATED PACKETS.

	Traffic Class	Previous Method	Proposed Method
Number of Generated Packets	RTC	360	360
	HPC	680	680
	MPC	1160	1160
	LPC	360	360
	Total Weighted	14040	14040
Number of Lost Packets	RTC	0	0
	HPC	232	197
	MPC	272	276
	LPC	18	2
	Total Weighted	2962	2682

4.2. Evaluation of congestion controller unit

In the previous part, the performance of the algorithm was evaluated for unbalanced instant traffics. However, the situation not always looks like this. In other words, the traffic variation is not always rapid rather it may occur gradually and might also remain constant for a period of time. There is also the possibility that after a gradual change, the network condition turns back to its previous status. In this case, it is in fact the rate control policy that tries to control the nodes' traffic rates by limiting the maximum transmission rates. By tuning this parameter for each node through the course of different scenarios, it is deemed to improve the performance of the network based on the perspective presented earlier. To see this, we devise three trials by exposing the network in

different probable conditions. The results are compared and contrasted, then. In each of the trials the geographical node priorities have been set to 1 for all nodes.

4.2.1. Heavy Traffic in a High Priority Node

It is fairly common for a WMSN traffic source to increase its packet generation rate for a period of time due to many reasons including triggering an event that begets permanent heavy communication in the geographical location where the node is placed. The traffic generation rate in such cases rate might even exceed the sink's maximum transmission rate, thus, as a compromise, the excited node should be assigned an acceptable amount of transmission rate with respect to other nodes.

In this trial, all the local traffic sources generate a traffic at speed of 10 packets/sec except for the HPC source located in M7 which follows the following paradigm; during the first 15 seconds of the simulation, the foregoing source rate increases linearly with a ramp of 10 packets/sec^2 after which it remains constant during the next 15 seconds and finally starts decreasing linearly with the same pace as the first stage (-10 packets/sec^2), until it reaches the inactivity level. The simulation runs for 50 seconds with a maximum transmission rate of 160 packets/sec per node. Fig. 12 demonstrates the average inter-arrival rate along with average transmission rate of M7 for both algorithms. According to the figure, it is clear how the new method easily adopts to the varying network conditions while the predecessor remains constant after it converged. Also, notice how the proposed method outperforms the previous method in having more sent packet rate. Another word under notation is that although in the proposed method the M7 maximum allowable transmission rate increases in proportion to the average inter-arrival packet rate, this increase is not without bound. This means that there is a compromise in how much the transmission rate is eligible to increase based on other existing nodes in the network. That is to say, if an increase in the link rate gives rise to a considerable consequent increase in the network weighted packet loss then it stops to become larger and larger. This fact is demonstrated clearly in the aforementioned figure where the average packet inter-arrival rate curve is much larger than the average transmission rate curve corresponding to the new method.

Fig. 13 depicts the plots of MPC and HPC queues' lengths for M7 during the time. Notice how the lower priority queue remains empty when the higher one is full. This is what we expect from the employed fuzzy controller while this is not true for the plot of non-fuzzy method at the bottom of the figure. It is also worthy of noting that this strategy decreases the delay corresponding to higher ranked queues significantly. This is portrayed in Fig. 14 where the average delay of the HPC queue for both methods are plotted.

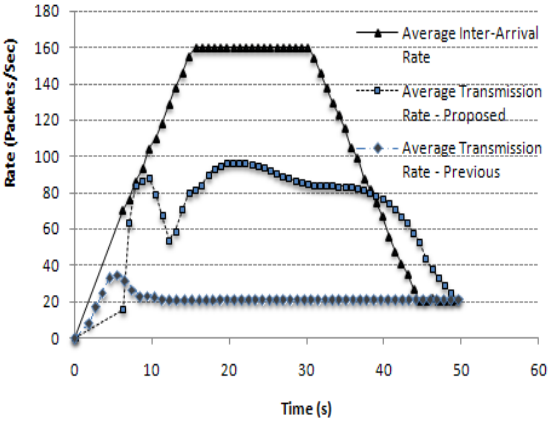

Fig. 12. Average transmission and inter arrival rate of M7 for the two algorithms.

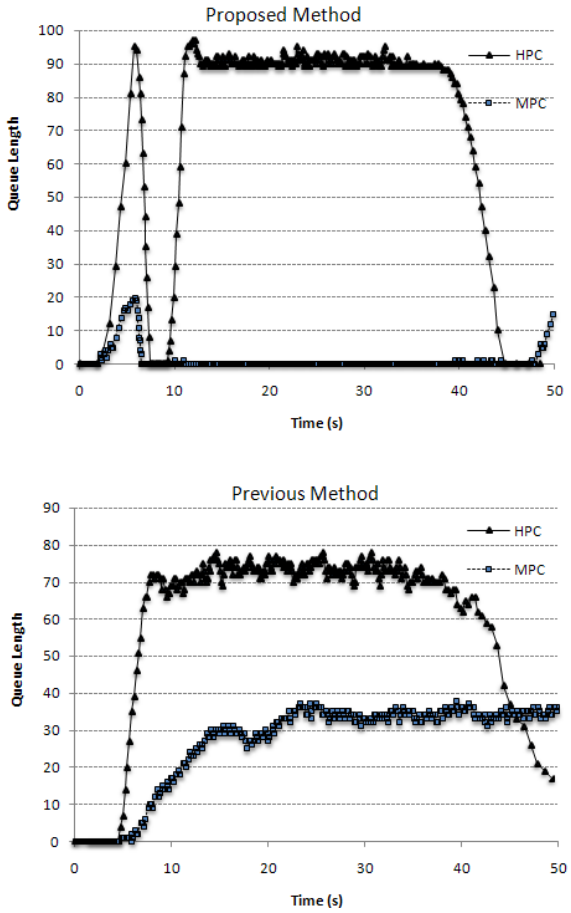

Fig. 13. Queue lengths for M7.

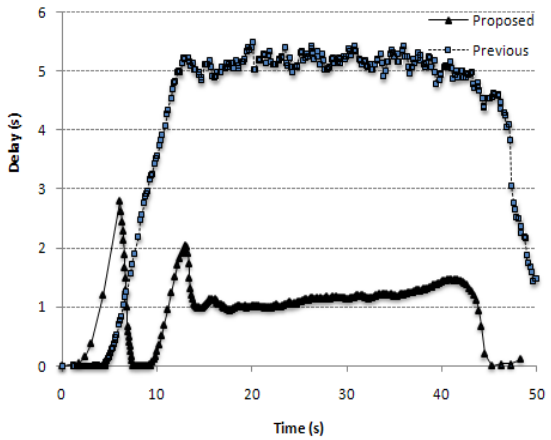

Fig. 14. M7 delay for the HPC class.

Fig. 15 depicts both the total weighted and unweighted loss probability for the two algorithms, during the time. According to the plots, not only the total weighted packet loss has been improved in the newly introduced method but also the unweighted metric appears in a better stance in regard of previous method; this is because of proper rate adjustment policy adopted in the new method based on the weighted packet loss metric which acts in a rather dynamic fashion.

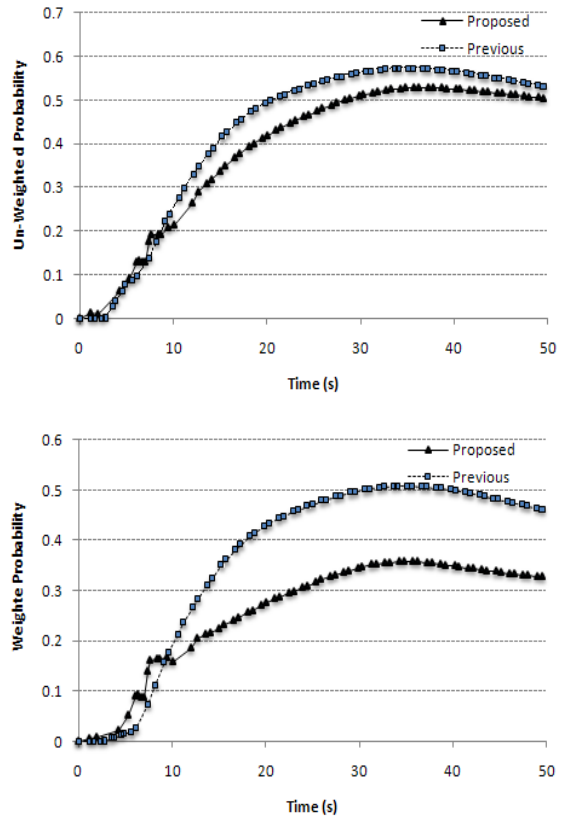

Fig. 15. Unweighted and weighted probability loss (top and bottom respectively).

Fig. 16 demonstrates the network throughput for each of the MPC and HPC classes. As expected, the throughput for the HPC class is considerably higher in the new method while this is not the case for the MPC class which is of lower priority.

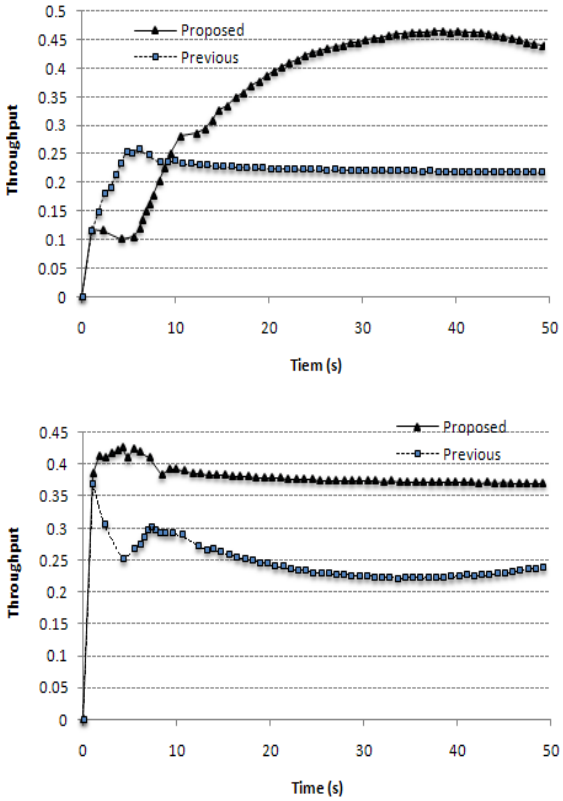

Fig. 16.Throughput discriminated base on type of classes (HPC top, MPC bottom).

Fig. 17 compares the average weighted packet loss metric for the two methods over the time. According to this figures due to an increase in the source rate, the average packet loss has also been increased, however, this is somewhat alleviated by the adopted rate adjustment mechanism proposed in the new method. To demonstrate how this is fulfilled, Fig. 18 shows maximum available transmission rates for both M1 and M2 during the time.

Based on this Fig., when the foregoing traffic source rate starts to increase, the rate adjustment unit changes the maximum allowable transmission rate. More specifically, this maximum threshold increases for M2 and decreases for M1 proportional to average weighted packet loss. The next point is that it takes some time for the proposed method to converge; this is one of the outstanding drawbacks of the new method. Finally, when the traffic rate corresponding to HPC decreases linearly these thresholds also return to their initial values for

better performance, which is implied by the figure. To sum up, Table 5 illustrates the number of generated and lost packets corresponding to each priority class after the simulation ends.

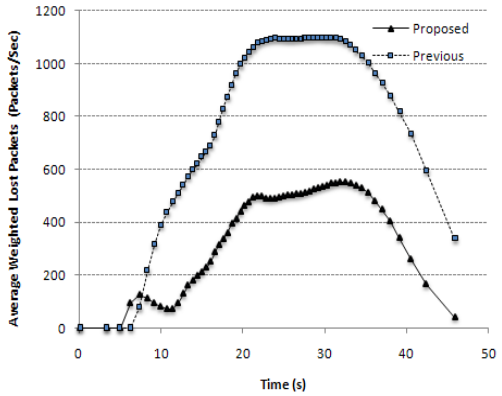

Fig. 17. Average weighted lost packets for M7.

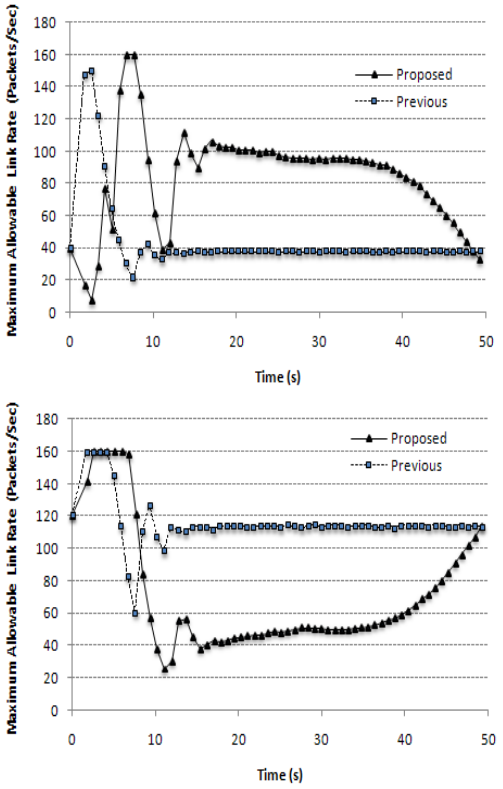

Fig. 18. Maximum allowable transmission rate for M2 (above) and M1 (bellow) nodes.

As already explained, the number of lost packets in HPC class is considerably lower in the new method as opposed to MPC class. Another important fact that should be underlined is that there exist a few packets in RTC class which are lost in the new method; this is because of higher convergence period of the proposed method comparing with the old one, as a trade off. This is also implied from Fig. 18 where the average transmission rates for the sensor nodes in the new method approach zero at the first moments of the simulation as a result of an undesired oscillation.

TABLE 5. OVERVIEW OF LOST/GENERATED PACKETS.

	Traffic Class	Previous Method	Proposed Method
Number of Generated Packets	RTC	1497	1497
	HPC	5550	5550
	MPC	3494	3494
	LPC	5550	5550
	Total Weighted	16091	78896
Number of Lost Packets	RTC	0	100
	HPC	3788	1878
	MPC	396	1213
	LPC	4353	4911
	Total Weighted	36241	25787

4.2.2. Heavy Traffic in Different Traffic Sources with Different Priorities

In the previous trial, the performance of the algorithm when confronting a traffic source with high generation rate, was evaluated and proved to perform efficiently in relation to the previous method. In this scenario, however, the network is exposed to a different situation where two different nodes located in different sub-trees of the network topology and with different levels of importance, gradually increase their packet generation rates. To see the robustness of the proposed method when dealing with such a situation, several metrics were plotted in the subsequent figures. As a first guess, we expect that the two excited nodes with different levels of priority should not receive the same level of attention and it may be required for the less important traffic source to have higher drop rate but allotting other nodes having more important packets to send higher maximum allowable transmission rates.

To describe the network configuration in this scenario, all the local traffic sources are generating packets at pace of 10 packets/sec except for the HPC traffic source in M7 and the LPC traffic source in M3 which both increase linearly with a ramp of 10 packets/sec^2 during the first 15 seconds of the simulation and remain constant for the next 5 seconds.

Fig. 19 depicts the average inter-arrival packet rate and average transmission rate of both M2 and M7 (the top and bottom plots), respectively, for the two algorithms. The simulation runs for 20 seconds. Fig. 20 shows the loss probability for M7 according to each method. Based on the figures, when the aforementioned source rate exceeds an endurable limit, it is the proposed method which handles the traffic more efficiently particularly after t=9sec. This is also illustrated in Fig. 21 based on another perspective, namely the average delay amount that packets need to wait until they get served in the HPC queue of node M7. Finally Fig. 22 depicts average weighted packet loss of M7 during the time.

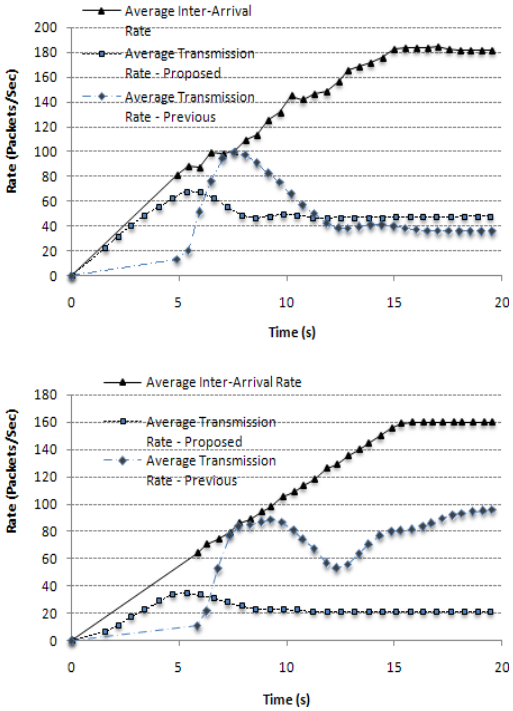

Fig. 19. Average transmission and inter-arrival rates of M3 (top) and M7 (bottom) for the two algorithms.

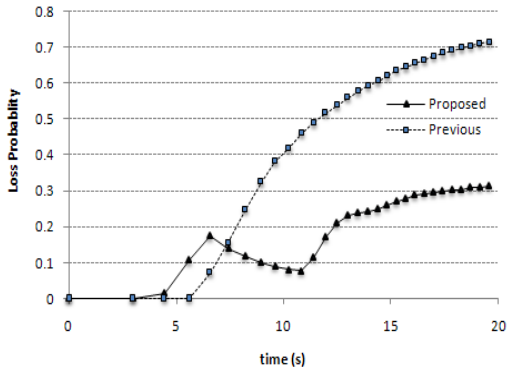

Fig. 20. M7 probability of loss

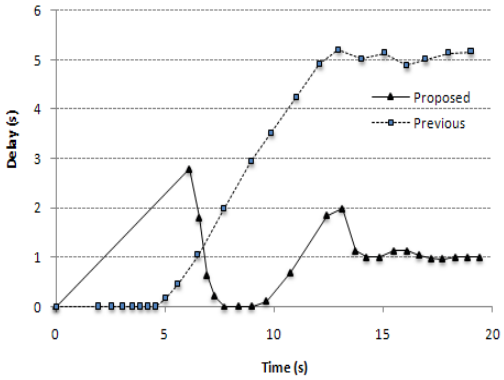

Fig. 21. M7 average delay in HPC queue.

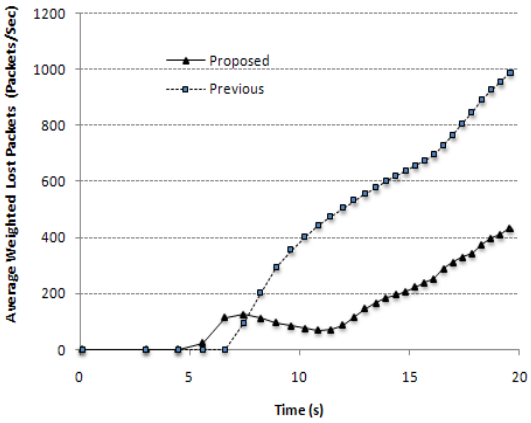

Fig. 22. M7 average weighted lost packets.

Table 6 illustrates the final results. Notice how the total weighted packet loss is substantially decreased in the new method.

TABLE 6. OVERVIEW OF LOST/GENERATED PACKETS.

	Traffic Class	Previous Method	Proposed Method
Number of Generated Packets	RTC	597	597
	HPC	2267	2267
	MPC	1394	1394
	LPC	2267	2267
	Total Weighted	31949	31949
Number of Lost Packets	RTC	0	13
	HPC	1470	747
	MPC	64	236
	LPC	1704	1741
	Total Weighted	13720	8791

4.2.3. Light Traffic Load in Network

In this last trial we highlight one of the main side-effects of the previous method when there is plenty of unused link capacity. The problem emerges by the fact that in the previous method each of the node's maximum transmission rate is periodically increased based on the available free bandwidth. In other words, if a node has nothing to send during a period of time then the maximum link rate for each child node will be increased blindly proportional to their priority. This finally gives rise to a situation where the maximum transmission rate corresponding to each of the foregoing nodes reaches its upper bound. Now consider a bursty traffic situation that comes afterwards at the same location (node). As the maximum allowable transmission rates have already set to the highest, they propagate the incoming burst to the upstream nodes and which begets a congestion that wastes considerable part of the bandwidth and energy of the system while this could be easily handled in the downstream nodes much earlier. The previous method is incapable of handling this phenomenon properly.

The network initial configuration in the described trial is as follows: the maximum value for nodes' transmission rates is opted to be 200 packets/sec (note that this will be tuned during the course of simulation) and all the local traffic sources generate 5 packets per second except for the HPC local traffic source in M7 which generates packets at a rate of 40 packets/sec. In other words, the total number of generated packets through the network $(13\times 5+40=105)$ roughly equals half of the sink's maximum transmission rate and consequently a plethora of the bandwidth remains unused at the beginning of the simulation. The simulation runs for 90 seconds.

In order to perform a quantitative comparison, we define the following measure indicating how well the maximum transmission rates are attributed to each node, as follows:

$$Wasted\ \ Bandwidth\ =\frac{1}{n}\sum_{n}(Max\ \ Rate-avgTsm)^2 \qquad (5)$$

This is an indication of how well the assigned capacity to each node is fits its transmission rate as it is not efficient to assign plethora of capacity for transmission of packets to a node that has nothing to send while there are many to send in a congested node. Fig. 23 plots the variation of this metric during the time. According to the figure, there is much more wasted bandwidth in the previous method than the proposed one which makes it evident that plenty of unused capacity has been assigned to different nodes.

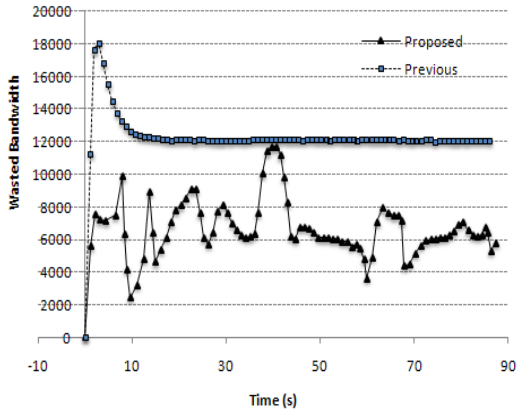

Fig. 23. Total wasted bandwidth.

Finally, Fig. 24 depicts the throughput discriminated based on packet priorities, however in this special case each plot appears to be in a better stance for the previous method. This does not contradict our assumption presented earlier rather a drastic increase in the maximum allowable transmission rates made the earlier method to outperform (however, at cost of putting the network vulnerable to congestion).

5. Conclusion and Future Work

In this article a new approach has been put forward that increases the quality of service in wireless multimedia sensor networks. The proposed method is designed to decrease the loss probability of packets in the network by managing the resources efficiently via considering the statistical features that are exclusive for the multimedia traffics, namely, the sudden and gradual traffic changes. We adopt two policies simultaneously to provide a synergy when dealing with these changes. The results are compared and contrasted with those of presented in [15] during different trials. The acquired results prove the superiority of the new method with respect to the competitor method. As every other method, the proposed model however, has its own drawbacks, the most important of which is the convergence time and the undesired oscillations.

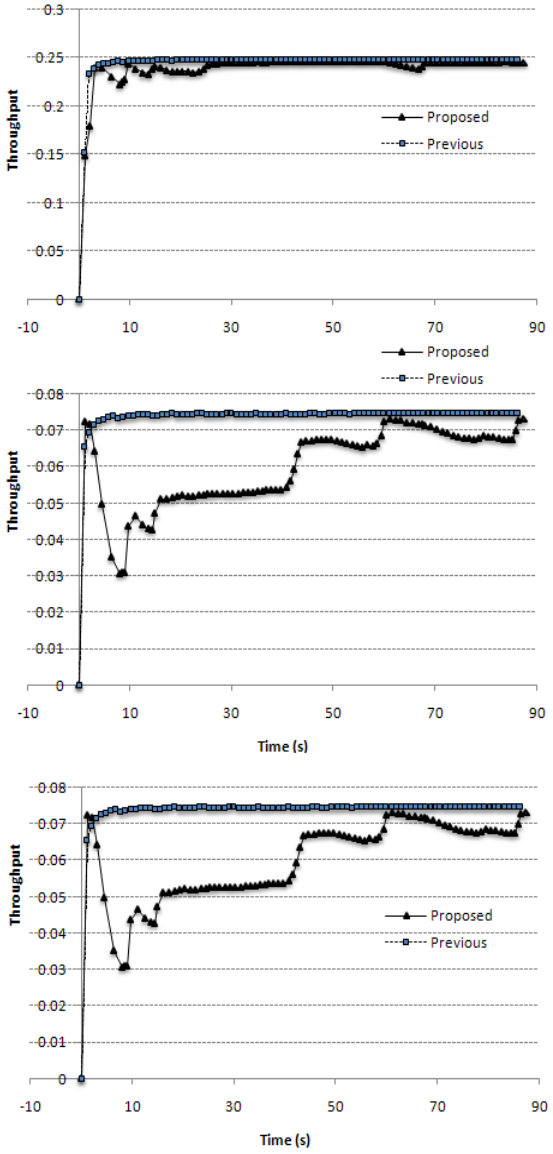

Fig. 24. Class throughput.

6. REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (4) (2002) 393– 422.

[2] I.F. Akyildiz, T. Melodia, T.R. Chowdhury, A survey on wireless multimedia sensor networks, Computer Networks 51 (2007) 921–960.

[3] V. Jacobson, Congestion Avoidance and Control, Symposium proceeding on communications architecture and protocols, (1988) 314-329.

[4] Yassine Hadjadj Aoul, Ahmaed Mehaoua, Charalabos Skianis, A Fuzzy Logic-Based AQM for Real-Time Traffic Over Internet, Computer Networks 51 (2007) 4617-4633.

[5] S. Floyd, K. Fall, Random early detection gateways for congestion avoidance, IEEE/ACM Transactions on Networking (August) (1993).

[6] Zadeh, L.: Fuzzy Sets, Information and Control, 8:338 – 353, (1965).

[7] Zargar, S.T., Yaghmaee, M.H.; Fard, A.M , Fuzzy Proactive Queue Management Technique, Annual India Conference, (2006).

[8] Tapio Frantti, Fuzzy Congestion Control In Packet Networks, Springer Berlin / Heidelberg, Volume 2/2005.

[9] B. Safaiezadeh, A.M. Rahmani, E. Mahdipour, A New Fuzzy Congestion Control in Computer Networks, International Conference on Future Computer and Communication, (2009) 314-318 .

[10] C.N. Nyirenda, D.S. Dawoud, "Self-Organization in a Particle Swarm Optimized Fuzzy Logic Con gestion Detection Mechanism for IP Networks", Submitted to Scientia Iranica, International Journal of Science and Technology.

[11] C. Wang, Member, K. Sohraby, M. Daneshmand, Y. Hu, Upstream congestion control in wireless sensor networks through cross-layer optimization, IEEE Journal on Selected Areas in Communications 25 (4) (2007) 786–795.

[12] C.-T. Ee, R. Bajcsy, Congestion control and fairness for many-to one routing in sensor networks, in: Proceedings of ACM Sensys, November 2004.

[13] C.-Y. Wan, S.B. Eisenman, A.T. Campbell, CODA: congestion detection and avoidance in sensor networks, in: Proceedings of ACM Sensys'03, Los Angeles, CA, November 5–7, 2003.

[14] H. Zhang et al., Reliable bursty convergecast in wireless sensor networks, in: Proceedings of ACM Mobihoc'05, Urbana-Champain, IL, May 25–28, 2005.

[15] Mohammad Hossein Yaghmaee a,*, Donald A. Adjeroh, Priority-based rate control for service differentiation and congestion control in wireless multimedia sensor networks, Computer Networks 53 (2009) 1798–1811.