

Adaptive Duty Cycle Scheduling Algorithm for Wireless Sensor Networks using Test Bed

M. Joseph Auxilius Jude, V.C. Diniesh, V. Yaknapriya

*Department of Electronics and Communication Engineering
School of Communication and Computer Science,
Kongu Engineering College, Perundurai, Erode, Tamil Nadu, India.
Jude2193@gmail.com, vcdiniesh@gmail.com, yaknapriya.v@rediffmail.com*

Abstract

Wireless Sensor Network consists of a group of sensors which has the capability to sense and transmit the real time information to the server. Scheduling in sensor gateway always remains a challenge as it needs to consider the optimal channel utilization of the sensor nodes. Most of the scheduling algorithm in the sensor gateway is based on node's duty cycle, i.e., active time, nap time and idle listening period of the detectors. The duty cycle scheduling is varied randomly based on the data arrival with FIFO queues as common consideration for diverse traffic loads. In this paper, a new queue based scheduling algorithm called Adaptive Duty Cycle Scheduling for embedded sensor networks is proposed for a diverse traffic condition to provide an efficient scheduling with reduced delay and energy consumption. The duty cycle for a sensor node is varied based on the arrival rate of data from sensor nodes and considering the queue length at the gateway. Extensive analyses are taken out under NS-2 simulation environment for SMAC and TMAC scheduling algorithm and their performance were mentioned for further enhancement. For the carrying out of proposed scheduling algorithm isense hardware module is utilized.

Keywords- Wireless Sensor Networks, Duty cycle, Energy Efficiency, Queue Management, Delay and Network Lifetime.

INTRODUCTION

Wireless Sensor Network (WSN) is mostly used in military and monitoring applications. Sensor network has the most number of sensors, which are spread across the environment. All these sensor nodes are sharing the same medium to

communicate, for providing efficient communication, scheduling algorithm is provided for WSN. These scheduling is based on the duty cycle of each sensor node. Duty cycle is the ratio of active period and the cycle time, cycle time is the duration of active time and the sleep time of sensor nodes. During active periods only each sensor node can transmit and receive the data. In the sleep time, the radio of each sensor node is turned off, during that time sensor node will not transmit data to other sensor nodes. Sensor nodes are battery powered. So to reduced energy consumption in sensor motes, sleep time is increased. If the active time is increased in the sensor mote, it consumes more energy. Also, idle listening time increases energy consumption. SMAC [1], [2], [3] protocol provides a fixed active time specified interval in active period is shared by the SYNC packet. SMAC provides the periodic, active and sleep time. The SYNC packet of SMAC has sleep-awake scheduling. By exchanging SYNC packet between the sensor motes, data transmission is achieved. If any RTS collision occurs then the sensor nodes are going into a sleep period until the active period is to occur. SMAC increases end-to-end delay.

In TMAC [3] algorithm the sensor node goes into a sleep period when no event occurred within the specified period in active time. TMAC reduces idle listening time. During the sleep time, if any data is transmitted to the sensor node. During that time, if the receiving node is in sleep time that data is buffered in the neighboring node's buffer. After the sleep time of that node, during active time the buffered data are transmitted from the nearby node to the receiver node. TMAC reduces energy consumption, but induces a delay because of an earlier end of active time. DSMAC [3] changes its duty cycle dynamically by increasing the active time. DSMAC increases its active time based on its application, also increases its energy consumption with increase in active time duration.

Related Work

SMAC provides better throughput when the duty cycle is high. In the sensor MAC protocol throughput of the sensor node is increased with the increment in the number of sensor nodes. At the certain point, the throughput of the SMAC protocol is saturated, and then it goes low, because of the RTS collision. It's active time can't be changed dynamically. If no data are to be transmitted or receive during the active period, more energy is wasted, because, until the end of an active time the radios of the sensor nodes are turned ON. It consumes more energy. If the data are detected at the end of the active time it will go to sleep time, then doesn't send the previously sensed data to the gateway it will send currently sensed data to the gateway. If in the case of measured data is critical, it produces a critical data loss. If the duty cycle of the SMAC has the low duty cycle, then it doesn't produce any critical data loss. There is no prioritization of data's in the sensor node having SMAC algorithm, for the understanding purpose of critical data loss here the prioritization is included in the SMAC algorithm. It will be compared to the proposed scheduling algorithm. In figure1, the right side log file shows the sensor node's output and the left side of the log file shows the gateway's output.

```

[2014-05-05 11:33:10.039] Booting Environment Module
Demo Application, id=13800
[2014-05-05 11:33:26.921] Medium priority data = 29 °C
[2014-05-05 11:33:27.944] Medium priority data = 29 °C
[2014-05-05 11:33:28.968] Medium priority data = 29 °C
[2014-05-05 11:33:29.925] SLEEP
[2014-05-05 11:33:33.990] WAKEUP
[2014-05-05 11:33:33.994] Medium priority data = 29 °C
[2014-05-05 11:33:34.055] Medium priority data = 29 °C
[2014-05-05 11:33:35.948] Medium priority data = 29 °C
[2014-05-05 11:33:36.060] SLEEP
[2014-05-05 11:33:40.065] WAKEUP
[2014-05-05 11:33:40.070] Medium priority data = 29 °C
[2014-05-05 11:33:41.076] Medium priority data = 29 °C
[2014-05-05 11:35:42.996] SLEEP
[2014-05-05 11:35:46.826] WAKEUP
[2014-05-05 11:35:46.997] Low priority data = 27 °C
[2014-05-05 11:35:47.813] High priority data = 32 °C
[2014-05-05 11:35:48.805] High priority data = 32 °C
[2014-05-05 11:35:48.840] High priority data = 32 °C
[2014-05-05 11:35:48.842] SLEEP
[2014-05-05 11:35:52.826] WAKEUP
[2014-05-05 11:35:52.996] Medium priority data = 29 °C
[2014-05-05 11:35:53.101] Medium priority data = 29 °C
[2014-05-05 11:35:53.106] Medium priority data = 29 °C
[2014-05-05 11:35:54.923] Medium priority data = 29 °C
[2014-05-05 11:36:21.273] SLEEP
[2014-05-05 11:40:11.226] WAKEUP
[2014-05-05 11:33:28.748] Queue is allocated with required
Size of 40 entries.
[2014-05-05 11:33:32.791] Current Queue Length is 1
[2014-05-05 11:33:32.796] Received Data is 29
[2014-05-05 11:33:33.796] Current Queue Length is 2
[2014-05-05 11:33:33.812] Received Data is 29
[2014-05-05 11:33:34.819] Current Queue Length is 3
[2014-05-05 11:33:34.823] Received Data is 29
[2014-05-05 11:33:38.825] Current Queue Length is 4
[2014-05-05 11:33:38.829] Received Data is 29
[2014-05-05 11:33:38.831] Current Queue Length is 5
[2014-05-05 11:33:38.840] Received Data is 29
[2014-05-05 11:33:39.837] Current Queue Length is 6
[2014-05-05 11:33:39.851] Received Data is 29
[2014-05-05 11:33:43.799] Current Queue Length is 7
[2014-05-05 11:33:43.813] Received Data is 29
[2014-05-05 11:34:43.819] Current Queue Length is 8
[2014-05-05 11:34:43.834] Received Data is 29
[2014-05-05 11:35:47.824] Current Queue Length is 9
[2014-05-05 11:35:47.834] Received Data is 27
[2014-05-05 11:35:47.846] Current Queue Length is 10
[2014-05-05 11:35:47.857] Received Data is 32

```

Figure1 Performance of SMAC Algorithm

From the above figure 9th sensed high priority data in the sensor node is not sent to the gateway node. The gateway node receives the measured data after its sleep time shown on the left side of the figure shown.

III. Proposed Algorithm

In the proposed algorithm the duty cycle of each sensor node is varied based on the arrival rate of sense data in gateway (queue). In this algorithm the queue is divided into three parts having two threshold values. Threshold values are fixed based on the application. The threshold value is the length of the split part of the queue. The every arrival data in the queue are compared with the threshold values. In this paper, the first threshold value is denoted as Th_1 and the next threshold value is denoted as Th_2 . Figure 2 shows the flow chart for adaptive duty cycle scheduling algorithm for sensor nodes and figure 3 shows the flow chart for adaptive duty cycle scheduling algorithm for gateway. This algorithm increases the sleep time of each sensor node based on the arrival rate of the data from a sensor node. The sensed data's are categorized as low priority data, medium priority data and high priority data. Here, the temperature sensor is used for the environmental monitoring. Based on the requirement the prioritizing is varied.

At the gateway, the queue length is compared with the arrival rate of the every detail. By increasing sleep time of a sensor node, arrival of data can be reduced. In the first case, if the queue length is less than the first thresh (Th_1) the then the sensor node sends all the sensed data's to the gateway. If the gateway reaches a first threshold (Th_1), then it will send this information to all the sensor nodes, which are in its network and then the sensor nodes stop sending its low priority data to gateway by going to sleep mode, it will send only the medium priority and high priority datas.

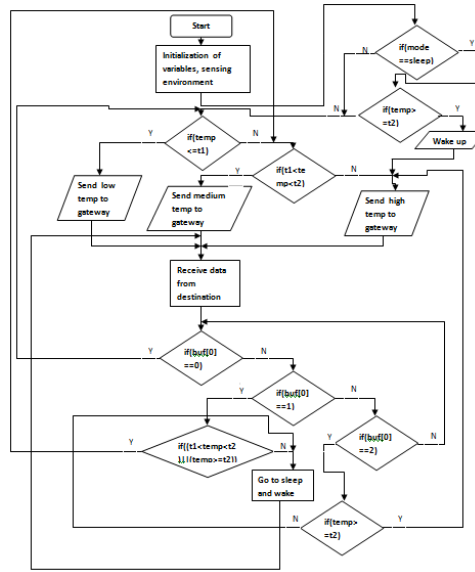


Figure2 Flow chart for queue based scheduling algorithm for sensor node

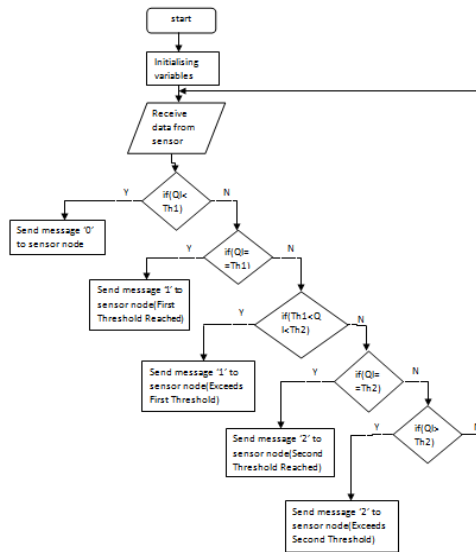


Figure 3 Flow chart for queue based scheduling algorithm for Gateway

In the second case, if the queue length is reached the second threshold (Th2), then the gateway sends this message to all sensor nodes, which are in its network and then the sensor nodes stop sending its low and medium priority data's to gateway and it will go to sleep mode after the sleep time it will go to active mode, then sense the environment changes then it will check the gateway condition and priority of sensed data then it will decide whether it will switch to sleep mode or continue with active mode.

While the sensor nodes are in sleep mode if any critical data (high priority data) sensed then the sensor nodes will go into active mode then follow the proposed scheduling algorithm. In this proposed algorithm during the sleep time if any critical data are detected, then the sensor node will go into an active mode, then send that critical data to the gateway, so that the critical data loss is reduced.

IV. Result and discussion

The figure 4 shows that at the initial state of sensor starts detection and categorizing the type of data, the queue length at the gateway will be less than that of first threshold. So, the sensor node will send all types of data, during that time the gateway module will send the '0' as a message to sensor node then only the sensor node knows that the queue doesn't get filled up to first threshold.

```

serialMonitorLogFile so - Notepad
[2014-04-29 12:02:42.426] Booting Environment Module
Demo Application, id=158d00
[2014-04-29 12:02:48.437] Low priority data = 25 °c
[2014-04-29 12:02:48.469] Received data is 0
[2014-04-29 12:02:53.473] Low priority data = 26 °c
[2014-04-29 12:02:53.489] Received data is 0
[2014-04-29 12:02:58.492] Low priority data = 26 °c
[2014-04-29 12:02:58.540] Received data is 0
[2014-04-29 12:03:03.545] Low priority data = 27 °c
[2014-04-29 12:03:03.577] Received data is 0
[2014-04-29 12:03:08.580] Medium priority data = 28 °c
[2014-04-29 12:03:08.596] Received data is 0
[2014-04-29 12:03:13.600] Medium priority data = 28 °c
[2014-04-29 12:03:13.609] Received data is 0
[2014-04-29 12:03:18.604] High priority data = 30 °c
[2014-04-29 12:03:18.618] Received data is 0
[2014-04-29 12:03:23.624] Low priority data = 27 °c
[2014-04-29 12:03:23.632] Received data is 0
[2014-04-29 12:03:28.627] Low priority data = 27 °c
[2014-04-29 12:03:28.653] Received data is 0
[2014-04-29 12:03:33.630] Low priority data = 27 °c
[2014-04-29 12:03:33.646] Received data is 0
[2014-04-29 12:03:38.653] Medium priority data = 28 °c
[2014-04-29 12:03:38.668] Received data is 1
[2014-04-29 12:03:38.675] data is 28
[2014-04-29 12:03:43.683] Medium priority data = 28 °c
[2014-04-29 12:03:43.698] Received data is 1
[2014-04-29 12:03:43.706] data is 28
[2014-04-29 12:03:48.689] Medium priority data = 28 °c
[2014-04-29 12:03:48.701] Received data is 1
[2014-04-29 12:03:48.713] data is 28
[2014-04-29 12:03:53.689] Medium priority data = 28 °c

serialMonitorLogFile De - Notepad
[2014-04-29 12:02:48.467] Current Queue Length is 1
[2014-04-29 12:02:48.475] Received Data is 25
[2014-04-29 12:02:53.482] Current Queue Length is 2
[2014-04-29 12:02:53.498] Received Data is 26
[2014-04-29 12:02:58.545] Current Queue Length is 3
[2014-04-29 12:02:58.550] Received Data is 26
[2014-04-29 12:03:03.578] Current Queue Length is 4
[2014-04-29 12:03:03.593] Received Data is 27
[2014-04-29 12:03:08.591] Current Queue Length is 5
[2014-04-29 12:03:08.603] Received Data is 28
[2014-04-29 12:03:13.609] Current Queue Length is 6
[2014-04-29 12:03:13.626] Received Data is 28
[2014-04-29 12:03:18.621] Current Queue Length is 7
[2014-04-29 12:03:18.630] Received Data is 30
[2014-04-29 12:03:23.621] Current Queue Length is 8
[2014-04-29 12:03:23.636] Received Data is 27
[2014-04-29 12:03:28.636] Current Queue Length is 9
[2014-04-29 12:03:28.644] Received Data is 27
[2014-04-29 12:03:33.652] Current Queue Length is 10
[2014-04-29 12:03:33.659] Received Data is 27
[2014-04-29 12:03:38.679] Current Queue Length is 11
[2014-04-29 12:03:38.685] Queue Reached Threshold 1
  
```

Figure 4 Output before reaching the first threshold

The figure 5 shows that after the gateway queue reached the first threshold, it will send messages to sensor nodes, and then the sensor nodes will send only the critical data and medium priority data to the gateway. If the low priority data is sensed by the sensor node, then the sensor node will go to sleep, then it will not send its previously sensed low priority data until gateway queue will reach second threshold. Sleep time calculated using the number of low priority data's detected in the sensor node after the first threshold is reached, for the every detection of high priority and medium priority data's the count of low priority data is to be reset.

The sleep time of the sensor node is varied dynamically based on the detection of low priority and medium priority data in sensor node. After reaching the first threshold, the sleep time calculation for every detection of low priority data is counted (L) and the sleep time is calculated as follows

$$\text{Sleep time} = (2\sqrt{L}) * 1000$$

where L is number of low priority data's detected

If any high priority or medium priority data are detected, then the count for low priority data is reset to 0. During sleep time if any high priority data (critical) are detected then it will go into active mode then it will continue the normal operation.

```

serialMonitorLogFile so - Notepad
[2014-04-29 11:49:14.273] Medium priority data = 28 °C
[2014-04-29 11:49:14.289] Received data is 1

[2014-04-29 11:49:19.292] Low priority data = 27 °C
[2014-04-29 11:49:19.302] SLEEP
[2014-04-29 11:49:22.282] WAKEUP

[2014-04-29 11:49:23.288] Medium priority data = 28 °C
[2014-04-29 11:49:23.304] Received data is 1

[2014-04-29 11:49:28.323] Medium priority data = 28 °C
[2014-04-29 11:49:28.338] Received data is 1

[2014-04-29 11:49:33.326] Medium priority data = 28 °C
[2014-04-29 11:49:33.338] Received data is 1

[2014-04-29 11:49:38.331] Medium priority data = 28 °C
[2014-04-29 11:49:38.346] Received data is 1

[2014-04-29 11:49:43.351] Low priority data = 27 °C
[2014-04-29 11:49:43.362] SLEEP
[2014-04-29 11:49:46.340] WAKEUP

[2014-04-29 11:49:48.354] Low priority data = 27 °C
[2014-04-29 11:49:48.366] SLEEP
[2014-04-29 11:49:49.361] WAKEUP

[2014-04-29 11:49:50.353] High priority data = 30 °C
[2014-04-29 11:49:50.369] Received data is 1

[2014-04-29 11:49:55.372] Low priority data = 27 °C
[2014-04-29 11:49:55.382] SLEEP
[2014-04-29 11:49:58.378] WAKEUP

[2014-04-29 11:49:59.370] Medium priority data = 29 °C
[2014-04-29 11:49:59.386] Received data is 1

[2014-04-29 11:50:04.389] Low priority data = 27 °C
[2014-04-29 11:50:04.399] SLEEP
[2014-04-29 11:50:07.394] WAKEUP

serialMonitorLogFile De - Notepad
[2014-04-29 11:49:14.287] Current Queue Length is 11
[2014-04-29 11:49:14.299] Queue Reached Threshold 1
[2014-04-29 11:49:14.308] Received Data is 28

[2014-04-29 11:49:23.309] Current Queue Length is 12
[2014-04-29 11:49:23.324] Queue Reached Threshold 1
[2014-04-29 11:49:23.331] Received Data is 28

[2014-04-29 11:49:28.324] Current Queue Length is 13
[2014-04-29 11:49:28.340] Queue Reached Threshold 1
[2014-04-29 11:49:28.347] Received Data is 28

[2014-04-29 11:49:33.338] Current Queue Length is 14
[2014-04-29 11:49:33.350] Queue Reached Threshold 1
[2014-04-29 11:49:33.358] Received Data is 28

[2014-04-29 11:49:38.352] Current Queue Length is 15
[2014-04-29 11:49:38.362] Queue Reached Threshold 1
[2014-04-29 11:49:38.370] Received Data is 28

[2014-04-29 11:49:50.376] Current Queue Length is 16
[2014-04-29 11:49:50.390] Queue Reached Threshold 1
[2014-04-29 11:49:50.396] Received Data is 30

[2014-04-29 11:49:59.395] Current Queue Length is 17
[2014-04-29 11:49:59.411] Queue Reached Threshold 1
[2014-04-29 11:49:59.422] Received Data is 29

[2014-04-29 11:50:31.432] Current Queue Length is 18
[2014-04-29 11:50:31.441] Queue Reached Threshold 1
[2014-04-29 11:50:31.447] Received Data is 30

[2014-04-29 11:51:03.487] Current Queue Length is 19
[2014-04-29 11:51:03.513] Queue Reached Threshold 1
[2014-04-29 11:51:03.518] Received Data is 28

[2014-04-29 11:51:15.515] Current Queue Length is 20
[2014-04-29 11:51:15.544] Queue Reached Threshold 1
[2014-04-29 11:51:15.557] Received Data is 28

[2014-04-29 11:51:20.568] Current Queue Length is 21
[2014-04-29 11:51:20.573] Queue Reached Threshold 2
[2014-04-29 11:51:20.578] Received Data is 28
  
```

Figure 5 First threshold

```

serialMonitorLogFile so - Notepad
[2014-04-29 12:37:21.355] High priority data = 30 °C
[2014-04-29 12:37:21.384] Received data is 2

[2014-04-29 12:37:26.375] Medium priority data = 28 °C
[2014-04-29 12:37:26.385] SLEEP
[2014-04-29 12:37:29.380] WAKEUP

[2014-04-29 12:37:31.379] Medium priority data = 28 °C
[2014-04-29 12:37:31.394] SLEEP
[2014-04-29 12:37:32.384] WAKEUP

[2014-04-29 12:37:33.392] Medium priority data = 28 °C
[2014-04-29 12:37:33.400] SLEEP
[2014-04-29 12:37:39.386] WAKEUP

[2014-04-29 12:37:40.394] Low priority data = 27 °C
[2014-04-29 12:37:40.403] SLEEP
[2014-04-29 12:37:43.398] WAKEUP

[2014-04-29 12:37:45.397] Low priority data = 27 °C
[2014-04-29 12:37:45.401] SLEEP
[2014-04-29 12:37:46.404] WAKEUP

[2014-04-29 12:37:47.411] Low priority data = 27 °C
[2014-04-29 12:37:47.419] SLEEP
[2014-04-29 12:37:53.405] WAKEUP

[2014-04-29 12:37:54.412] Low priority data = 27 °C
[2014-04-29 12:37:54.420] SLEEP
[2014-04-29 12:38:01.416] WAKEUP

[2014-04-29 12:38:02.423] Low priority data = 27 °C
[2014-04-29 12:38:02.432] SLEEP
[2014-04-29 12:38:06.419] WAKEUP

[2014-04-29 12:38:07.440] High priority data = 30 °C
[2014-04-29 12:38:07.456] Received data is 2

[2014-04-29 12:38:12.442] High priority data = 31 °C
[2014-04-29 12:38:12.454] Received data is 2

serialMonitorLogFile De - Notepad
[2014-04-29 12:37:21.382] Current Queue Length is 32
[2014-04-29 12:37:21.396] Queue Reached Threshold 2
[2014-04-29 12:37:21.407] Received Data is 30

[2014-04-29 12:38:07.441] Current Queue Length is 33
[2014-04-29 12:38:07.456] Queue Reached Threshold 2
[2014-04-29 12:38:07.465] Received Data is 30

[2014-04-29 12:38:12.454] Current Queue Length is 34
[2014-04-29 12:38:12.470] Queue Reached Threshold 2
[2014-04-29 12:38:12.477] Received Data is 31

[2014-04-29 12:38:17.468] Current Queue Length is 35
[2014-04-29 12:38:17.479] Queue Reached Threshold 2
[2014-04-29 12:38:17.484] Received Data is 32

[2014-04-29 12:38:22.481] Current Queue Length is 36
[2014-04-29 12:38:22.493] Queue Reached Threshold 2
[2014-04-29 12:38:22.503] Received Data is 32

[2014-04-29 12:38:27.494] Current Queue Length is 37
[2014-04-29 12:38:27.502] Queue Reached Threshold 2
[2014-04-29 12:38:27.509] Received Data is 30

[2014-04-29 12:38:32.507] Current Queue Length is 38
[2014-04-29 12:38:32.519] Queue Reached Threshold 2
[2014-04-29 12:38:32.536] Received Data is 30

[2014-04-29 12:39:00.559] Current Queue Length is 39
[2014-04-29 12:39:00.576] Queue Reached Threshold 2
[2014-04-29 12:39:00.590] Received Data is 30

[2014-04-29 12:39:05.572] FREEING QUEUE

[2014-04-29 12:39:05.582] Current Queue Length is 1
[2014-04-29 12:39:05.591] Received Data is 30

[2014-04-29 12:39:10.571] Current Queue Length is 2
  
```

Figure 6 Second threshold

The figure 6 shows that after queue length in the gateway reached second threshold then the gateway queue send message to sensor node which is associated with that gateway. After receiving this message, then the sensor node will send only the critical data, if medium priority data will arrive at the sensor node, then that sensor node will not be sent that data to the gateway and also it will go to sleep mode. During this sleep mode if any critical data will arrive then it will go into active mode send that data to the gateway.

The figure 6 shows that the sensor node goes to sleep mode if it has the low and medium priority data's. During that time if any critical data doesn't arrive, then the sensor node will go into active mode after the fixed sleep time. If critical data arrives, then the sensor node will go into active mode during sleep mode. Due to that the critical data loss will not occur and also the delay in accessing critical data will be reduced. After reaching the second threshold in the gateway, if the low priority data and medium priority data is detected then it will count in the calculation of sleep time. Here, the sleep time for the sensor node is varied dynamically. To calculate the sleep time if low priority data are detected then the above mention equation is used. If the medium priority data are detected the following formula is used for the calculation

$$\text{Sleep time} = (2\sqrt{M}) * 1000$$

where M is the number of medium priority data's detected.

If high priority data are detected, then the count for low priority (L) and medium priority (M) is reset to '0'. During sleep time if any High priority data (critical) are detected then it will go into active mode then it will continue the normal operation.

Conclusion

Existing MAC protocols have more drawbacks and results in more packet loss and no proper scheduling algorithm for sensor network. Hence, in order increase WSN lifetime and compatible with traffic, new generation queue based scheduling called Adaptive Duty Cycle Scheduling for Embedded Sensor Networks is designed for traffic and sensor gateway side with reduced delay and low energy consumption. The proposed Adaptive duty cycle scheduling is implemented in wireless sensor motes. For the scheduling operation, it uses the parameters of gateway and sensor motes. From the gateway it gets the queue length, from the sensor node, it gets the number of low priority and medium priority data's. So, that it reduces the congestion at the gateway and reduces the delay for accessing high priority data. When compared with SMAC algorithm the proposed algorithm reduces the critical data loss. Because, SMAC has the fixed active time and fixed sleep time duration. During sleep time sensor node will not sense and send or receive data's. But, in the case of proposed algorithm the sleep time and the active time will be varied dynamically based on the sensed data. During sleep time if any critical data are detected, the sensor node goes into an active mode so that no critical data loss will occur and reduce the delay in accessing high priority data.

REFERENCES

- [1] Changes Suh, Zeeshan Hameed Mir, Young-Bae Ko (2008) “Design and Implementation of Enhanced IEEE 802.15.4 For Supporting Multimedia Service in Wireless Sensor Networks”, Springer Journal of Computer Networks; vol 52; No 2;pp 2568-2581
- [2] Christophe J. Merlin, Heinzelman B (2010), “Duty Cycle Control for Low-Power-Listening MAC Protocols”, IEEE Transactions on Mobile Computing; vol. 9; No. 11
- [3] Feng Liu, Chi-Ying Tsui and Ying (2010), “Sleep Scheduling for Lifetime Maximization of Wireless Sensor Networks”, IEEE Transaction on Wireless Communications; vol 9; no.7
- [4] Heejung Byun, Junglok Yu (2013), “Adaptive Duty Control with Queue Management in Wireless Sensor Networks”, IEEE Transaction on Mobile Computing; vol.12; no.6
- [5] Heejung Byun, Soogook Son, Jungmin So (2013), “Queue management based duty cycle control for end-to-end delay guarantees in wireless sensor networks”, Springer Journal of Wireless Networks; vol.19, no 5, pp. 1349–1360
- [6] Heidemann W.Ye, J, Estrin D (2004), “Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Network”, IEEE Transaction on Networking; vol.12; No.3; pp 493-506
- [7] Joohwan Kim, Xiaojun Lin (2010), “Minimizing Delay and Maximizing Lifetime for Wireless Sensor Networks with Anycast”, IEEE Transaction on Networking; Vol.12; No.3; pp 493-506
- [8] Ke Liu. Nael, Abu-Ghazaleh, Kyoung-Don and Kang (2010), “Exploiting Slack Time for Just-in-Time Scheduling in Wireless Sensor Networks,” Springer Journal of Real-Time-System; vol 45; pp 1-25
- [9] Paolo Bronte, Prashant Pillai, Vince Cook W.C, Stefano Chessa (2007), “Wireless Sensor Networks: A Survey on the State of The Art and The 802.15.4 and ZigBee Standards”, Springer Journal of Computer Communications; vol 30; pp 1655-1695
- [10] Peng Guo, Tao Jian, Qian and Zhan Kui Zhan (2012),“Sleep Scheduling for Critical Event Monitoring in Wireless Sensor Networks”, IEEE Transaction on Parallel and Distributed Systems, vol 23; No.345-352.
- [11] Wee Lum TAN, Wing Cheong Lau, OnChing Yue (2012), “Performance Analysis of an Adaptive, Energy-Efficient MAC Protocol for Wireless Sensor Networks”, Springer Journal of Parallel Distribution; vol 72; pp 504-514.
- [12] Zhao, Ma C.V, Miao M (2010), “An energy efficient and low-latency MAC protocol with Adaptive Scheduling for multihop wireless sensor network”, Springer Journal of Computer Communications vol 33; pp 1452–1461.
- [13] <http://www.coalesenses.com/index.php/products/solutions/isense-modules/>
- [14] <http://www.coalesenses.com/index.php/products/solutions/isense-software/ishell/>