Variable Workload Scheduling With Increased Throughput in Wireless Sensor Networks

Mrs. S. Jeevitha
M.E Software Engineering, Mount Zion College of Engineering and Technology, Pudukkottai, Tamilnadu, India

Abstract

Wireless sensor networks are appealing to researchers due to their wide range of application potential in areas such as target detection and tracking, environmental monitoring, industrial process monitoring, and tactical systems. Existing approaches are insufficient for developers to determine whether system's requirements concern the communication latency, bandwidth utilization, reliability, or energy consumption. Since the data of sensors are expressed in form of expressive queries, the performance of query services should be increased especially for high data rate sensor network applications. In this paper it is decided to propose Real Time Conflict-free Query Scheduling (RTCQS), an enhanced transmission scheduling technique for real time queries in wireless sensor networks. The query preemption algorithm is used to achieve high throughput while scheduling without preemption is used for the queries that are able to execute concurrent.

***Keywords*—**Wireless Sensor Networks, throughput, Scheduling, Interference-Communication graph, Inter release time

I. INTRODUCTION

Sensor Networks have emerged as one of the dominant technology trends of decade (2000-2010) that has potential usage in defense and scientific applications [1]. A Wireless Sensor Network' (WSN) can be described as a network of sensors which communicate with each other wirelessly. These sensors may be installed in an unattended environment with limited computation and sensing capabilities. A typical node in the WSN consists of a sensor, embedded processor, moderate amount of memory and transmitter/receiver circuitry. These sensor nodes are normally battery powered and they coordinate among themselves to perform a common task [2]. The ability to communicate not only allows information and control to be communicated across the network of nodes, but nodes to cooperate in performing more complex tasks, like statistical sampling, data aggregation, and system health and status monitoring [3,4].

A. Wireless Sensor Network Protocol Stack

The communication architecture of a wireless sensor network, implemented by a protocol stack, is shown in Fig.1 [5] built out of 5 layers and 3 cross-layer planes. Medium access protocols reside in the Data Link Layer, which itself not only is responsible for a fair distribution of resources, but also for providing data stream multiplexing, frame detection and error control. Parts of these goals are accomplished by MAC protocols.

The main duties of the medium access protocol are firstly to assist the construction of a network infrastructure, and secondly to control the medium access, so that all sensor nodes in the network have equal access to the resources and use them as efficiently as possible[5]. In order to be able to implement time-slotted MAC protocols, clock synchronization between the sensor nodes in the wireless network is an important requirement. This task can be rendered possible by the Sensor Management Protocol on the Application Layer.

B. Real Time Applications

Initial applications supported by WSNs were mostly in environment monitoring, such as temperature monitoring for a specific area, house alarming, and so on. The main objectives in such applications only involved simple data processing. Energy consumption needed to be considered for specific applications, so little attention was taken on data delivery and reliability related issues [6] [7] and [8]. WSNs have been extended and their design have been advanced to support more hard design and complex applications, such as security, military, fire detection and health care related applications. In these applications, data delivery and reliability must be taken as important parameters in addition to energy efficiency, because data must be collected from the sources of events and be forwarded to the sink in real time with high reliability, otherwise the application will be useless. Some most important design factors for protocols in WSN that need to be considered while designing and deploying energy efficient MAC protocols for any applications are the following: network topology, type of antenna and clustering related issues.

Sensor Networks serve many diverse applications starting from the research on Great Duck Island (GDI) [1] for monitoring the maine to high data rate real time structural health monitoring, the query services requires the improvement of performance in terms of query throughput and latency. For meeting this communication need, Real Time Conflict free query scheduling (RTCQS) is designed as a WSN protocol which includes the common properties of WSN query services. It provides the effective

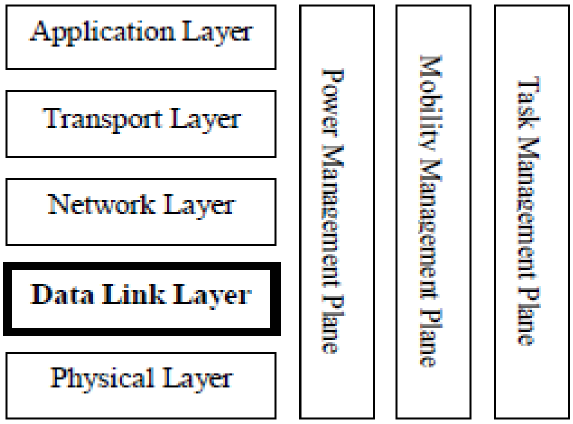


Fig.1 Protocol Stack of WSN

prioritization between traffic classes meeting various deadlines and supports high throughput since they generate a very high workload. It is proposed to have the following properties such as: 1) Adapting the query schedule for every workload change 2) Achieving a high throughput and low latency 3) Ability to work on resource constrain devices 4) Predictability in query rate and power consumption

II. RELATED WORKS

A common Medium Access Control (MAC) paradigm used in wireless network is Carrier Sense Multiple Access (CSMA) [9]. It is simple and flexible not requiring any complex infrastructure, clock synchronization or global topology. Any CSMA based medium access scheme has two important components, the listening mechanism and the backoff scheme. But it doesn’t work beyond one hop causing problem called hidden terminal problem which may leads to degradation of throughput especially in high data rate sensor application.

Time division multiple access (TDMA) is a potential candidate for WSNs. It can be defined as the process of allocating time slots to the nodes or links between each pair of neighboring nodes, to ensure collision free channel access. It is most attractive for high data rate sensor networks because it is energy-efficient and may provide higher throughput than CSMA/CA protocols under heavy load. The two types of scheduling carried out in TDMA are node scheduling where the scheduler assigns slots to nodes and link scheduling where slots are assigned to links through which pairs of nodes communicate. Though it is efficient than CSMA, it has many disadvantages making less suitable for use of sensor network [10]. First, finding an efficient time schedule in a scalable fashion is not trivial. Second TDMA needs clock synchronization which may incur high energy overhead because it requires frequent message exchanges. Third, it is very expensive to handle when sensor networks undergo frequent topology changes because of time-varying channel conditions, physical environmental changes, and battery outage and node failures. Fourth, during low contention TDMA gives much lower channel utilization and higher delays than CSMA In addition it is not suitable for real time applications with variable workloads as it maintains an explicit schedule for transmission. Thus the MAC scheme for sensor network should include a variant of TDMA [11].

Z-MAC (Zebra MAC) is a hybrid MAC scheme that combines the ascents of TDMA and CSMA for sensor networks while offsetting their descents [12]. The main feature of Z-MAC is its adaptability to the level of contention in the network. It is robust to dynamic topology changes and time synchronization failures commonly occurring in sensor networks. Z-MAC uses CSMA as the baseline MAC scheme, but uses a TDMA schedule as a hint to enhance contention resolution. In Z-MAC, a time slot assignment is performed at the time of deployment therefore a higher overhead is incurred at the beginning. As the maximum slot number is broadcasted in Z-MAC, security issue is to be taken into account.

Distributed Randomized TDMA Scheduling (DRAND) is fully distributed, efficient scalable channel scheduling algorithm [13]. It is the first scalable implementation of RAND which is a famous centralized channel scheduling scheme. DRAND calculates a TDMA schedule in time linear to the maximum node degree in form of time slot. After the slot assignment, each node reuses its assigned slot periodically in every predetermined period, called frame. A node assigned to a time slot acts as an owner of that slot and the others be the non-owners of that slot. It gives a chance of being more than one owner per slot. It is useful in scheduling protocols such as Z-MAC, FDMA, CDMA etc.,

TRaffic-Adaptive Medium Access (TRAMA) protocol provides energy-efficient conflict free channel access in wireless sensor networks [14]. Energy efficiency is attained by using the transmission schedules that avoid collisions of data packets at the receivers having nodes switch to low power radio mode when there is no data packets intended for those nodes. It supports for unicast, broadcast and multicast traffic and more adaptive for sensor network and monitoring applications. But it is not suited for delay sensitive transmission.

Several protocols aim at supporting real-time communication in multi-hop networks by proposing real-time transmission scheduling for robots [15]. Both protocols may assume that at least one robot has complete knowledge of the robots’ positions and network topology. Though these protocols are suited for small teams of robots, they are not suitable for queries in multi-hop WSNs.

In Dynamic Conflict Free Query Scheduling (DCQS) [16], a transmission scheduling technique for WSN queries is designed to support variable workloads and to exploit specific communication patterns and temporal properties of queries in WSNs. This allows DCQS to achieve high throughput. But it does not support query prioritization or real-time communication.

III. SYSTEM ARCHITECTURE

RTCQS adopts a variant of node/link scheduling called query scheduling in contrast to earlier TDMA protocols. In query scheduling, the time slots are assigned to transmission on specific communication. This helps to achieve the high throughput, low power consumption, scalability and to adopt topology changes.

Fig.2 shows the system architecture of RTCQS. It mainly aims at executing the real time queries in efficient manner. For this, it is designed to use the data aggregation functions such as packet merging, data compression or stationary function.

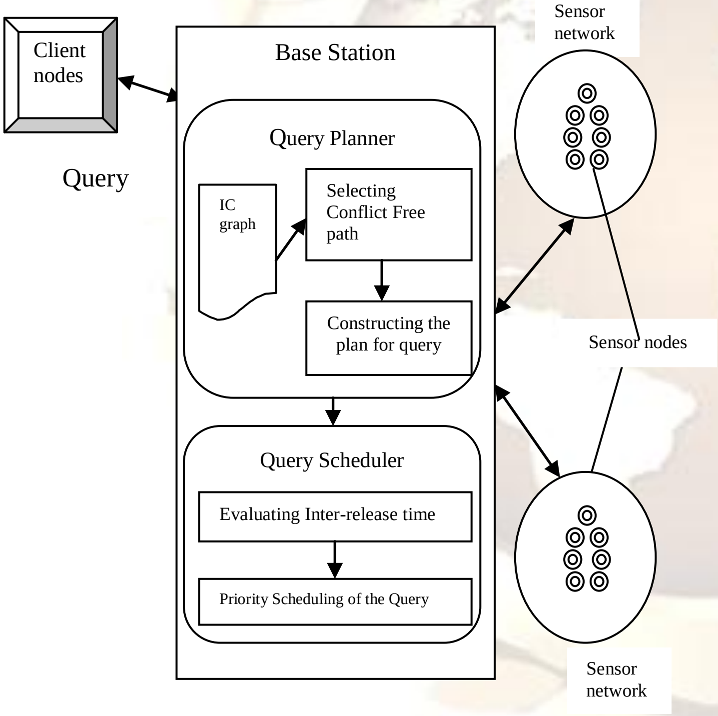


Fig.2 System architecture of RTCQS

It may use the packet merging [18] as a default aggregation function to reduce the energy consumption and transmits workload. The two parameters influencing the performance of query services are throughput and real time capacity. The former is summation of execution frequencies of all the queries while the latter is the maximum throughput for which the query service does not drop packets and meets the query deadlines.

A. Components.

The two main components used in the system are query planner and query scheduler. The query planner decides the transmissions for each query and scheduler for scheduling the corresponding plans.

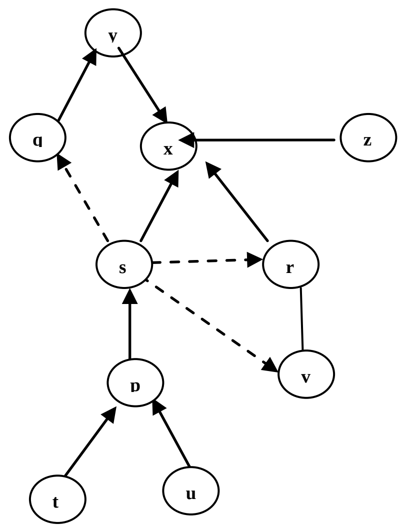
1) Query Planner:

The query planner is responsible for constructing the plans which is the sequential order of transmission steps for executing a query instance. Each step should consist of set of transmissions which should be free from both the primary and secondary conflict. Primary conflict is said to occur when one node transmits and receives at the same time slot or receives more than one transmission destined to it at same time slot. Secondary conflict occurs when an intended receiver of particular transmission is also within the transmission range of another transmission intended for other nodes. Conflict free path can be determined by planning the transmission with help of Interference Communication (IC) graph.

Let G (V,E) be the IC graph where V are the set of vertices representing sensor nodes and E are the set of edges represents the communication edges and interference edges. The link between the nodes for packet transmission is termed as the Communication edges and the link which interrupts the any communication in time is Interference edge. For example, PQ and RS are said to be conflict free $(\overrightarrow{PQ} \parallel \overrightarrow{RS})$ if PS and RS are not the edges and P, Q, R, S are distinct. The realistic method for constructing the IC graph is Radio Interference Detection (RID) based on Receiver Signal Strength (RSS) [17]. An example IC graph is shown in Fig.3

The plan with the steps 0 to 6 is shown for the example IC graph in Fig.4. In each step the transmissions assigned are conflict free. For example, in step 2 the nodes v and p may transmit simultaneously as their transmission $\overrightarrow{vr}$ and $\overrightarrow{ps}$ are conflict free i.e, they do not conflict with each other $(\overrightarrow{vr} \parallel \overrightarrow{ps})$. It provides the transmission to satisfy the precedence constraints such that t and u transmits before its parent p.

A node may communicate with its one hop neighbour to construct a plan at local. If the plan involves more number of hops then the plan is formulated by a node n with higher priority and sends PRequest packet. Upon receiving PRequest from node n it may check for its own one hop neighbour. If no such node exists the receiver may respond with PFeedback packet along with its local plan. In next stage node n disseminates its local plan to its one hop neighbour using PSend. On receiving the PSend packet, it acknowledge with PCommit packet. In order to balance the uneven workload demands across the



nodes, the planner would assign multiple steps for these nodes to transmit to achieve their workload demand. The minimized transmission plan may results in reduced latency.

		Senders						
		x	y	r	z	s	q	p
Steps	0							u
	1							t
	2			v		p		
	3	s						
	4	z						
	5	r						
	6	y						

Fig. 4 Constructed Plan for IC graph in Fig.3

2) Query Scheduler:

As the data transmission in WSN makes use of expressive queries, query scheduling is used to avoid the wastage of time when the nodes or links are idle for a period of time. Instead of assigning the time slot for node or links, slot should be assigned for the query. The slot is a period of time allotted for workload on demand. Scheduling should ensure that all the steps executed are conflict free with the relative order being preserved.

Each node is having a local scheduler to schedule the transmission steps. The Scheduler may contain the start time, period of queries, plan's length and min. inter-release time. The min. inter-release time is defined as the minimum time step between the subsequent instances of query to be executed.

The algorithm for scheduling without Pre-emption is used when the two instances of a query is executed concurrently. The scheduler uses two different queues called wait and execute queue. The instance of query waiting to be executed are stored in wait queue but are not being executed and those instance to be executed is placed in execute queue. The min. inter-release time is taken as ∆ between any two instances. When no instances are executed and the step distance between the head of wait queue and tail of execute queue is larger than ∆, an instance of query is started.

When an instance starts, it is moved from wait queue to execute queue. Being simple and efficient, it is feasible on resource constraint devices. The operation of determining the starting time of query instance takes time of O (1). The algorithm for query pre-emption is used to pre-empt the instance of query that conflict with the execution of higher priority instance. The algorithm is shown using Algorithm1.

```
begin
new instance i is released
wait = wait ∪ i

    begin
    start a new slot s
    for each i in wait
    if (may-continue(i) = true) then
    continue (i)
    for each i ∪ execute
    run (i)
    end

end

continue (i):
execute = execute ∪ i;
wait  = wait – i add instances to all occurConflict

preempt(S):
execute = execute − S;
wait = wait ∪  S;
remove i from all occurConflict

may-continue(i):
if (occurConflict = null) then
return true
if  (i  has  higher  priority  all  instances  in
occurConflict)
preempt(occurConflict);
return true
return false

run(i):
determine if node should send/recv
i = i + 1
if  i = query length then execute = execute – i
occurConflict = occurConflict – i
occurConflict = occurConflict  ∪ i
```

Algorithm1. Query Pre emption Algorithm

When the query instance is pre-empted, it determines whether the query instance to be pre-empted is conflict with existing instance. It maintains a wait and execute queue by the priority of instances. When a new instance is released for execution, it should be stored in wait queue. It starts/continues an instance i in two cases: (1) If the next step s+1 may be executed concurrently with all instances in the execute queue without conflict, it starts/resumes it. Then it checks if occurConflict (set having instance conflict with instance at execution) is empty. When an instance is started or continued, it is moved from the wait to execute. (2) i is also started/resumed if it has higher priority than all the instances in occurConflict. For i to be executed without conflict, all instances in occurConflict must be preempted. When an instance is preempted, it is moved from the wait to the execute queue and it is removed from all occurConflict sets.

IV. PERFORMANCE EVALUATION

The performance of DRAND is compared with the performance of the proposed RTCQS system in terms of query throughput and query latency. DRAND is a state of art protocol which does not include any interference relationship among nodes. Hence the results of it may produce collision. RTCQS enforces the conflict free transmission and achieves a high throughput and reduced latency.

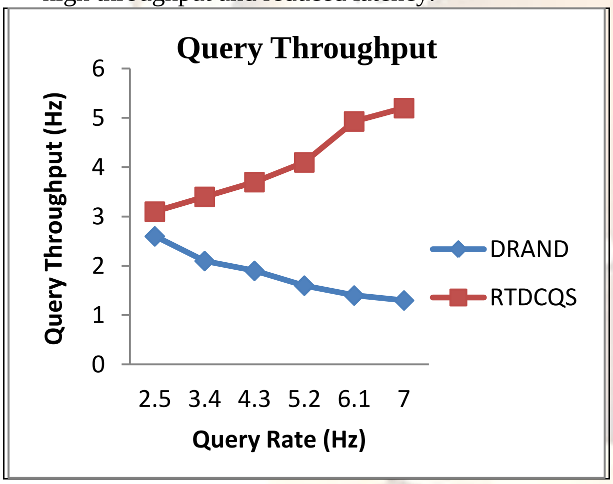


Fig. 5 Performance comparison using query Throughput for DRAND and RTCQS

A) Performance Metric. The query throughput and query latency of the proposed system is compared with that of DRAND which are defined as follows:

1) Query Throughput: It is the number of query instances completed per second. It is expressed in Hertz (Hz)

2) Query Latency: It is the response time for every query instance after sending the query request. It is represented by seconds (s).

B) Results.

Figure 5 shows the performance comparison of DRAND and RTCQS in terms of throughput. Clearly shown, RTCQS achieves the maximum throughput of 5.2Hz which is about 57% higher than DRAND. From this it is concluded that fair allocation of slots to nodes is unsuitable for WSN.

Figure 6 shows the performance comparison of DRAND and RTCQS in terms of latency. Even though the query rate

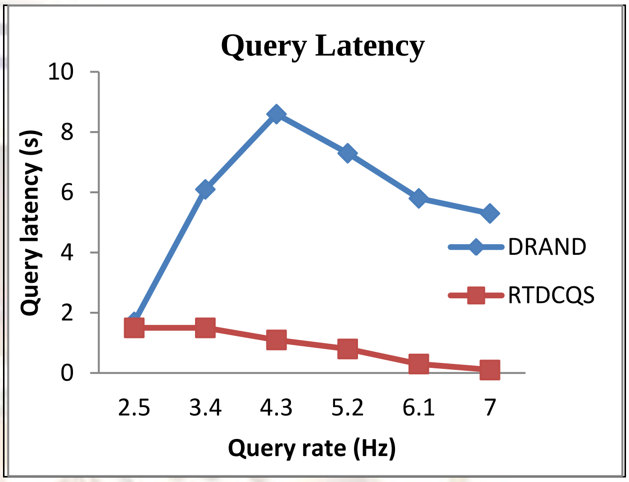


Fig. 6 Performance comparison using query latency for DRAND and RTCQS

is low, the RTCQS performs significantly better latency on comparing DRAND. For example, when query rate is 3.4, DRAND has the latency of 6.4 in contrast to RTCQS with latency of 1.5 which is about 77% lower than DRAND. The long latency period for DRAND is due to increased waiting duration a node to transmit entire frame to its parent.

V. CONCLUSION

The scheduling technique RTCQS is specifically designed for real time query services in wireless sensor networks. With the query planner, query latency is reduced by constructing conflict free transmission plans based on the precedence constraints. By the query scheduler throughput is improved by over-lapping the transmissions of multiple query instances concurrently. Scheduler makes use of both preemption and non preemption technique to enable the real time applications. Scheduler without the preemption controls only the start of an instance; In contrast, a preemption technique may preempt an instance to allow a higher priority instance to execute when the two cannot be executed concurrently. Thus RTCQS has low runtime overhead and limited memory requirements making it suitable for resource constrained devices and produce high throughput. In future, the proposed technique can be applied in WSN with higher topology change to have better performance.

ACKNOWLEDGMENT

Author express thanks to their friends and family members for encouraging her in good manner for completing this work.

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci: A Survey on Sensor Networks, IEEE Communication Magazine, pp. 102-114 (August 2002).

[2] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J.Anderson, "Wireless Sensor Networks for Habitat Monitoring,"Proc. First ACM Int'l Workshop Wireless Sensor Networks and Applications (WSNA), 2002.

[3] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. "Instrumenting the world with wireless sensornetworks". In International Conference on Acoustics,Speech, May 2001.

[4] D. Estrin, R. Govindan, J. S. Heidemann, and S. Kumar." Next century challenges: Scalable coordination in sensor networks". In Mobile Computing and Networking, pages 263–270, 1999.

[5] M. A. Rahman and S. Hussain, "Optimizing a data routing protocol's influence on communication connectivity," in IEEE Consumer Communications and Networking Conference (CCNC'07), Jan. 11-13 2007.

[6] P. Suriyachai, U. Roedig, and A. Scott, "A survey of mac protocols for mission-critical applications in wireless sensor networks," Communications Surveys Tutorials, IEEE, vol. PP, no. 99, pp. 1 –25, 2011.

[7] E. Felemban, C.-G. Lee, and E. Ekici, "Mmspeed: multipath multi-speed protocol for qos guarantee of reliability and. timeliness in wireless sensor networks," Mobile Computing, IEEE Transactions on, vol. 5, no. 6, pp. 738 – 754, June 2006.

[8] M. Soyturk and D. Altilar, "Reliable real-time data acquisition for rapidly deployable mission-critical wireless sensor networks," in INFOCOM Workshops 2008, IEEE, April 2008, pp. 1 –6.

[9] A. Woo, D. Culler, "A transmission control scheme for media access in sensor networks", Proceedings of ACM MobiCom'01, Rome, Italy, July 2001, pp. 221–235.

[10] W. Ye, J. Heidemann, and D. Estrin. "Medium access control with coordinated adaptive sleeping for wireless sensor networks", IEEE/ACM Trans. Netw., 12(3):493{506, 2004.

[11] K. Sohrabi, J. Gao, V. Ailawadhi, G.J. Pottie, "Protocols for self-organization of a wireless sensor network", IEEE Personal Communications, October 2000, pp. 16

[12] I. Rhee, A. Warrier, M. Aia, and J. Min, "Z-MAC: A Hybrid MAC for Wireless Sensor Networks," Proc. Third Int'l Conf. Embedded Networked Sensor Systems (SenSys), 2005.–27.

[13] I. Rhee, A. Warrior, J. Min, and L. Xu, "DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad Hoc Networks," Proc. ACM MobiHoc, 2006.

[14] V. Rajendran, K. Obraczka, and J.J. Garcia-Luna-Aceves, "Energy- Efficient Collision-Free Medium Access Control for Wireless Sensor Networks," Proc. First Int'l Conf. Embedded Networked Sensor Systems (SenSys), 2003.

[15] A. Koubaa, M. Alves, and E. Tovar, "i-GAME: an implicit GTS allocation mechanism in IEEE 802.15.4 for time-sensitive wireless sensor networks," in Euromicro Conference on Real-Time Systems, 2006.

[16] O. Chipara, C. Lu, J. Stankovic, and G. Roman, "Dynamic conflict-free transmission scheduling for sensor network queries," IEEE Transactionson Mobile Computing, vol. 10, no. 5, 2011.

[17] G. Zhou, T. He, J.A. Stankovic, and T.F. Abdelzaher, "RID: Radio Interference Detection in Wireless Sensor Networks," Proc. IEEE

[18] J. Polastre, J. Hui, P. Levis, J. Zhao, D. Culler, S. Shenker, and I. Stoica, "A unifying link abstraction for wireless sensor networks," in Embedded Networked Sensor Systems, 2005.