INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

SCHEDULING AGGREGATE QUERIES IN MULTIHOP SENSOR NETWORKS

Prabakaran.K[1], Mathan kumar.M[2], Dr. S. Uma[3]., B.E., M.S., PhD.

[1]*P.G. Scholar, PG Department of CSE, Hindusthan Institute of Technology,Coimbatore-32*
[2] *Assistant Professor, PG Department of CSE, Hindusthan Institute of Technology,Coimbatore-32s*
prabakarankprabu@gmail.com

[3]*Professor and Head PG Department of Computer Science and Engineering,*
Hindusthan Institute of Technology Coimbatore.

Abstract

This paper proposes Real-Time Query Scheduling (RTQS), a novel approach to conflict-free transmission scheduling for real-time queries in wireless sensor networks. First, we show that there is an inherent trade-off between prioritization and throughput in conflict-free query scheduling. We then present three new real-time scheduling algorithms. The non-pre-emptive query scheduling algorithm achieves high throughput while introducing priority inversions. The pre-emptive query scheduling algorithm eliminates priority inversion at the cost of reduced throughput. The slack stealing query scheduling algorithm combines the benefits of pre-emptive and non-pre-emptive scheduling by improving the throughput while meeting query deadlines. We propose Dynamic Conflict-free Query Scheduling (DCQS), a novel scheduling technique for queries in wireless sensor networks. In contrast to earlier TDMA protocols designed for general-purpose networks and workloads, DCQS is specifically designed for query services supporting in-network data aggregation. DCQS has several important features. First, it optimizes the query performance and energy efficiency by exploiting the temporal properties and precedence constraints introduced by data aggregation. Second, it can efficiently adapt to dynamic workloads and rate changes without explicitly reconstructing the transmission schedule.

1. Introduction

Many WSN applications use query services to periodically collect data from sensors to a base station. In this paper, we propose Real-Time Query Scheduling (RTQS), a transmission scheduling approach for real-time queries in WSNs. To meet this challenge, we present a set of new real-time query scheduling algorithms and associated schedulability analysis, which bridge the gap between WSNs and real-time scheduling theories. Our scheduling algorithms exploit the unique characteristics of WSN queries including many-to-one communication, in-network aggregation and periodic timing properties. This paper makes four contributions: First, we show through analysis and experiments that query scheduling has an inherent trade-off between prioritization and throughput. Second, we developed three scheduling algorithms: (1) The non-pre-emptive query scheduling algorithm achieves high throughput at the cost of some priority inversions. (2) The pre-emptive query scheduling algorithm achieves good prioritization by eliminating priority inversions. (3) The slack stealing scheduling algorithm combines the advantages of pre-emptive and non-pre-emptive scheduling algorithms by improving the throughput while meeting all query deadlines. Third, we derive latency upper bounds for each scheduling algorithm. This enables us to guarantee that the admitted queries meet their deadlines. Our analysis enables query services to handle overload conditions through online admission and rate control. Finally, we provide simulations that demonstrate the advantages of RTQS over contention-based and TDMA-based protocols in term of both real-time performance and throughput.

Our Main Contributions: Due to unique challenges for PAQS, we need to propose a novel design of scheduling protocols to orchestrate both the real-time job scheduling and in-network aggregation for answering a given set of queries. This is one of the two main contributions of our work. For a set of periodic data aggregation queries, we design a family of routing, node- and packet-level scheduling protocols under various wireless interference models such that each query can be satisfied (the sink node can receive all the data for each query), within a bounded end-to-end delay. The main idea for our protocol design is to split the sensor network spatially and temporally and then to find a schedule that makes efficient and careful use of resources. We prove theoretically that our protocol can achieve a total load that is at least a constant fraction of the optimum load; and at the same time, for each query, the delay is at most a small constant factor of the minimum delay by which any protocol can achieve.

Our second main contribution lies in schedulability test schemes that can test whether a given set of queries can be satisfied using any possible method. We propose necessary conditions for schedulability (summarized in Theorem 6), such that if a set of queries does not satisfy the conditions, we can determine immediately that the WSN is overloaded with query tasks (or the total request rate of all queries exceeds the 'capacity' of the network). We also propose sufficient conditions for schedulability of a set of periodic aggregation queries (summarized in Theorem 5) based on our protocol design. The gap between the proposed sufficient conditions and necessary conditions is proved to be a constant. This implies that the proposed sufficient conditions can achieve a utilization that is at least a constant fraction of the optimum utilization for schedulability. In addition to theoretical analysis, we conduct extensive simulation studies of our protocol design and schedulability test, the result of which corroborate our theoretical analysis.

2 Related Works

TDMA protocols can provide predictable packet latencies and achieve higher throughput than contention-based protocols under heavy load. The IEEE 802.15.4 standard for WSNs has a reservation mechanism for providing predictable delays in single hop networks. A more flexible slot reservation mechanism is proposed where slots are allocated based on delay or bandwidth requirements. Two recent papers proposed real-time communication protocols for robots. Both protocols assume that at least one robot has complete knowledge of the robots' positions and/or network topology. While the protocols may work well for small teams of robots, they are not suitable for queries in large-scale WSNs. Implicit EDF provide prioritization in a single-hop cell. The protocol supports multi-hop communication by assigning different frequencies to cells with potential conflicts. However, the protocol does not provide prioritization for transmitting packets across cells. In contrast, RTQS provides prioritization even in large multi-hop networks without requiring multiple frequencies. Two recent protocols that support real-time flows in WSNs have been proposed. In a scheduling based solution is proposed to support voice streaming over realtime flows. The real-time chains protocol extends a contention-based scheme called Black Burst to support packet prioritization. However, these protocols only support real-time flows involving only one or a few data sources. In contrast, RTQS is optimized for real-time queries that collect sensor data from many sources. In earlier work we proposed DCQS [6], a TDMA protocol that achieves high throughput by exploiting explicit query information provided by the query service. However, DCQS does not support query prioritization or realtime communication, which is the focus of this paper.

2.1 Real-time Scheduling

Two representative classes of well-studied real-time scheduling algorithms are rate-monotonic (RM) and EDF scheduling. The class of RM algorithms assigned static-priorities to queries on the basis of the cycle duration of the jobs. Liu and Layland presented a RM algorithm in a single processor, and the first sufficient condition for schedulability of a set of queries. On the other hand, EDF is a dynamic scheduling algorithm. EDF and its several extensions were proposed to guarantee the end-to-end delay of packets, e.g., EDF with traffic shaper that can regulate the distorted traffic from the EDF scheduler to deal with the bursty traffic. Unfortunately, using optimal traffic shaper is in general infeasible and introduces additional packet delays. Another approach, such as deadline-curve based EDF (DC-EDF) [42], or similar one, is to judiciously adjust the local deadlines of packets at a node,

Vol.2 Issue.5, Pg.: 52-57
May 2014

based on the traffic load and/or the end-to-end deadlines. DC-EDF can guarantee end-to end delay performances and provide a schedulable region as large as that of RC-EDF.

3 Query Model

RTQS assumes a common query model in which source nodes produce data reports periodically. This model fits many applications that gather data from the environment at user specified rates. A query l is characterized by the following parameters: a set of sources, a function for in network aggregation, the start time _l, the query period Pl, the query deadline Dl, and a static priority. A new query instance is released in the beginning of each period to gather data from the WSN. We use IL, u to refer to the uth instance of query l whose release time is rl, u = l +u · Pl. For brevity, in the remainder of the paper we will refer to a query instance simply as an instance. The priority of an instance is given by the priority of its query. If two instances have the same query priority, the instance with the earliest release time has higher priority. For each query instance a node i need Wl [i] slots to transmit its (aggregated) data report to its parent. A query service works as follows: a user issues a query to a sensor network through a base station, which disseminates the query parameters to all nodes. The query service maintains a routing tree rooted at the base station. The query service supports in-network data aggregation. Accordingly, each non-leaf node waits to receive the data reports from its children, produces a new data report by aggregating its data with the children's data reports, and then sends it to its parent. During the lifetime of the application the user may issue new queries, delete queries, or change the period or priority of existing queries. RTQS is designed to support such workload dynamics efficiently.

Data Aggregation Tree Construction

$T = (V_T, E_T), V_T = V, E_T \leftarrow \emptyset$
    ▷ /* Connect black nodes layer by layer */
for $i \leftarrow 1$ to $l-1$ **do**
    for all black nodes $v \in BLACK_{i+1}$ **do**
        Find its parent $p(v)$ in BFS tree
        Color $p(v)$ blue
        Find $p(v)$'s dominator $d_{p(v)}$ in $BLACK_i \cup BLACK_{i-1}$
        Add an edge between $p(v), v$ to E_T
        Add an edge between $d_{p(v)}, p(v)$ to E_T
    end for
end for
    ▷ /* Connect remaining white nodes */
for all remaining white nodes u **do**
    Find u's dominator d_u
    Add an edge between u, d_u to E_T
end for
return T

In this phase, we first construct a breadth first search tree for the network and divide all nodes into layers (where the 0-th layer is the base station and the 1st layer is its neighbors). We then form a maximal independent set BLACK layer by layer (Algorithm 1) as follows. Starting from the 1st layer, we pick up a maximal independent set and mark these nodes black. We then move on to the 2nd layer and pick up a maximal independent set and mark these nodes black again. Note that the black nodes of the 2nd layer also need to be independent of those of the 1st layer. We repeat this process until all layers have been worked on. Those who are not marked black are marked white at last.

4 Simulations

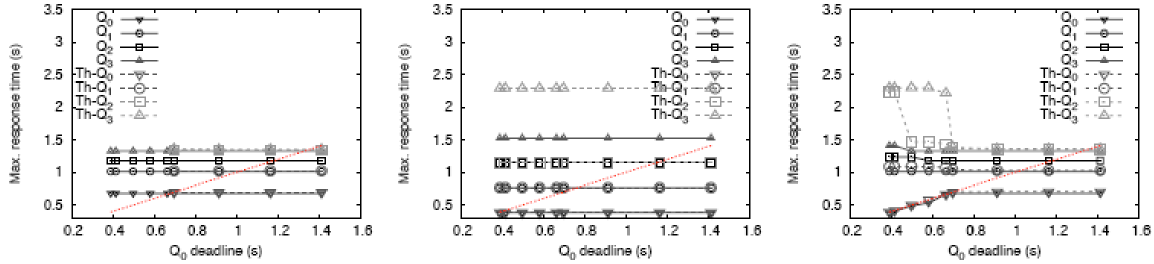
We implemented RTQS in NS2. Since we are interested in supporting high data rate applications such as structural health monitoring we configured our simulator according to the 802.11b settings having a bandwidth of 2Mbps. This is reasonable since several real-world structural health monitoring systems use 802.11b interfaces to meet their bandwidth requirements. An overview of these deployments may be found in [15]. At the physical layer a two-ray propagation model is used. We model interference according to the Signal-to-Interference-plus-Noise-Ratio (SINR) model, according to which a packet is received correctly if its reception strength divided by the sum of the reception strengths of all other concurrent packet transmissions is greater than a threshold (10 dbm in our simulations). In the beginning of the simulation, the IC graph is constructed using the method described. The node closest to the center of the topology is selected as the base station. The base station initiates the construction of the routing tree by flooding setup requests. A node may receive multiple setup requests from different nodes. The node selects as its parent the node that has the best link quality indicator among those with smaller depth than itself. We determined the slot size as follows. We assume that a node samples its accelerometer at 100Hz and buffers 50 16-bit data points before transmitting its data report to its parent. To reduce the number of transmissions, data merging is employed: a node waits to receive the data reports from its children and merges their readings with its own in a single data report which it sends to its parent. In our experiments, the maximum number of descendants of any node is 20, so the maximum size of a data report containing 16-bit measurements is 2KB. Accordingly, we set the slot size to 8.3ms, which is large enough to transmit 2KB of data. In our simulations, all queries are executed according to the same plan as every node sends its data report in a slot. For comparison we consider three baselines: 802.11e, DCQS[6] and DRAND[18]. We did not use 802.15.4 as a baseline, since the standard is designed for low data rate applications and hence is unsuitable for our target high data rate applications. 802.11e is a representative contention based protocol that supports prioritization in wireless networks.

In our simulations we use the Enhanced Distributed Channel Access (EDCA) function of 802.11e since it is designed for ad hoc networks. EDCA prioritizes packets using different values for the initial back-off, initial contension window, and maximum contention window of the CSMA/CA protocol. We configured these parameters according to their defaults in 802.11e. We used the 802.11e NS2 module from [11]. DRAND is a recently proposed TDMA protocol. DCQS is a query scheduling algorithm that constructs TDMA schedules to execute queries. However, neither DCQS nor DRAND support prioritization or real-time transmission scheduling.

4.1 COMPARISON OF RTQS ALGORITHMS

In this subsection we compare the performance of all RTQS algorithms and validate their response time analysis. We consider four queries Q0, Q1, Q2, and Q3 in decreasing order of priority. The ratios of their periods Q0:Q1:Q2:Q3 is 1.0:1.2:2.2:3.2. In this experiment, we fix the rates of the queries and vary the deadline of the highest priority query. Figs. 1(a) - 1(c) show the maximum response times of NQS, PQS, and SQS, respectively. For clarity, only Q0's deadline is plotted since in this experiment the other queries always meet their deadlines. PQS meets Q0's deadline when it is 0.39s. In contrast, NQS meets its deadline only when Q0's deadline is bigger than 0.69s. NQS misses Q0's deadline when it is tight due to the priority inversion under non-pre-emptive scheduling. This indicates that NQS is unsuitable for high priority queries with tight deadlines. Interestingly, under SQS, the response time of Q0 changes depending on its deadline (Fig. 7(c)). As the deadline becomes tighter, the response time of Q0 also decreases and remains below the deadline.

We also see an increase in the response times of the lower priority queries asQ0's deadline is decreased. This is because as Q0's deadline decreases the lower priority queries may steal less slack from Q0. This shows that SQS adapts effectively based on query deadlines. Moreover, note that SQS provides smaller latencies for the lower priority instances than PQS. This is because SQS has a higher throughput than PQS since it uses pre-emption only when it is necessary for meeting packet deadlines. In all experiments, the measured response times of all RTQS algorithms are lower than the worst-case response times derived using our analysis. Hence, our analysis is correct. The difference between the simulation results and the theoretical bounds are expected because the analysis is based on worst-case arrival patterns which do not always occur in simulations.



(a) NQS max. response time (b) PQS max. response time (c) SQS max. response time

Figure 1: IJRCAR SQS adapts to different deadlines

5 CONCLUSIONS

High data rate real-time queries are important to many sensor network applications. This paper proposes RTQS, a novel transmission scheduling approach designed real-time queries in WSNs. We observe that there exists a trade-off between throughput and prioritization under conflict-free query scheduling. We then present the design and schedulability analysis of three new real-time scheduling algorithms for prioritized transmission scheduling. NQS achieves high throughput at the cost of priority inversion, while PQS eliminates priority inversion at the cost of query throughput. SQS combines the advantages of NQS and PQS to achieve high query throughput while meeting query deadlines. NS2 simulations results demonstrate that both NQS and PQS achieve significantly better real-time performance than representative contention-based and TDMA protocols. Moreover, SQS can maintain desirable real-time performance by adapting to deadlines.

REFERENCES

[1] G.-S. Ahn, A. T. Campbell, A. Veres, and L.-H. Sun. Swan: service differentiation in stateless wireless ad hoc networks. In INFOCOM ’02.

[2] A. N. Audsley, A. Burns, M. Richardson, and K. Tindell. Applying new scheduling theory to static priority preemptive scheduling. Software Engineering Journal, 1993.

[3] B. D. Bui, R. Pellizzoni, M. Caccamo, C. F. Cheah, and A. Tzakis. Soft real-time chains for multi-hop wireless adhoc networks. RTAS ’07.

[4] M. Caccamo, L. Y. Zhang, L. Sha, and G. Buttazzo. An implicit prioritized access protocol for wireless sensor networks. In RTSS, 2002.

[5] O. Chipara, C. Lu, and C.-G. Roman. Real-time query scheduling for wireless sensor networks. Technical Report WUCSE-2007-10, Washington University in St. Louis.

[6] O. Chipara, C. Lu, and J. A. Stankovich. Dynamic conflictfree query scheduling for wireless sensor networks. In ICNP, 2006.

[7] T. Facchinetti, L. Almeida, G. C. Buttazzo, and C. Marchini. Real-time resource reservation protocol for wireless mobile ad hoc networks. In RTSS ’04.

[8] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly. Distributed multi-hop scheduling and medium access with delay and throughput constraints. In MobiCom ’01.

[9] K. Karenos, V. Kalogeraki, and S. Krishnamurthy. A rate control framework for supporting multiple classes of traffic in sensor networks. In RTSS, 2005.

[10] ALICHERRY,M., BHATIA, R., AND LI, L. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In ACM MobiCom (2005), pp. 58–72.

[11] BEAVER, J., SHARAF,M., LABRINIDIS,A., AND CHRYSANTHIS, P. Location-Aware Routing for Data Aggregation in Sensor Networks. In Geosensor Networks (2004), pp. 189–209.

[12] BOURAS, C., AND SEVASTI, A. A delay-based analytical provisioning model for a QoS-enabled service. In IEEE ICC (2006), pp. 766–771.

[13] BRAR, G., BLOUGH, D., AND SANTI, P. Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. In ACM MobiCom (2006), pp. 2–13.

[14] CHEN, X., HU, X., AND ZHU, J. Minimum Data Aggregation Time Problem in Wireless Sensor Networks. In LNCS (2005), pp. 133–142.

[15] CHIPARA, O., LU, C., AND ROMAN, G. Real-time query scheduling for wireless sensor networks. In IEEE RTSS (2007), pp. 389–399.

[16] CHIPARA, O., LU, C., AND STANKOVIC, J. Dynamic conflictfree query scheduling for wireless sensor networks. In IEEE ICNP (2006), pp. 321–331.

[17] CONSIDINE, J., LI, F., KOLLIOS, G., AND BYERS, J. Approximate aggregation techniques for sensor databases. In IEEE ICDE, pp. 449–460.