Competent Scheduling for Intermittent Aggregation Queries in Multihop Sensor Networks

B.Madhavilatha[1], A.L.Sreenivasulu[2]
[1]PG student, Department of CSE, Intell Engineering College, Anantapur, AP, INDIA,

[2]Assistant Professor, Department of CSE, Intell Engineering College, Anantapur, AP, INDIA,

Abstract:
Recent advances in low-power computing and communication technologies have given rise to the proliferation of wireless sensor networks having low cost sensor nodes with limited processing capacity and battery power. Wireless sensor networks can be used in a wide range of applications such as industrial process monitoring and control, surveillance of critical areas and structures, and target tracking. Scheduling is the process of deciding how to commit resources between a variety of possible tasks. However in this paper we examine periodic query scheduling for data aggregation with minimum delay under interference models. For a set of periodic aggregation queries, every query which is having its own period and the subset of source nodes which contains the data. For that we propose a family of efficient and effective real time scheduling protocols that can answer every job of each query task within a relative delay. Scheduling protocol contains the phases routing, transmission plan constructions, node activity scheduling, and packet scheduling. Based on this protocol design, we further propose schedulability test schemes to efficiently and effectively test whether, for a set of queries ,every query job can be finished within a finite delay.
Keywords:
Aggregation, delay, periodic, query scheduling, schedulability.

1. Introduction:

A wireless sensor network (wsn) in its simplest form can be defined as a network of (possibly low-size and low-complex) devices denoted as nodes that can sense the environment and communication the information gathered from the monitored field through wireless links; the data is forwarded, possibly via multiple hops relaying, to a sink that can use it locally, or is connected to other networks (e.g., the Internet) through a gateway.
Given a WSN consisting of a set of sensor nodes and the control center (or sink node), the sink will issue to the network a set Q of periodic aggregation queries. For each query Qi € Q , the sink may be interested in data only from a certain region, and thus only a subset of nodes will generate data to satisfy the query. We call these nodes source nodes. For each period of each query, the sink expects to receive the corresponding (possibly aggregated) data from the source nodes. For a given wireless interference model (we do not restrict ourself to a specific interference model), this objective is to jointly design a routing tree for each query and an interference-free schedule of activities for all nodes devise a way to protect confidentiality and integrity of data from sensors and queries (modeled as range queries) from the sink. (i.e., when to transmit and which packet to transmit) such that for each query , every job can be answered within a finite delay .We note the problem as Periodic Aggregation Query Scheduling (PAQS). Due to unique challenges for PAQS, we will propose a novel design of scheduling protocols to orchestrate both the real-time job scheduling and in-network aggregation for answering given queries. 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. Our main idea is to split the sensor network spatially and temporally and find a schedule that makes efficient and careful use of resources. We prove that our protocol can achieve a total load that is at least a constant fraction of the optimum load. 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 test whether a given set of periodic aggregation queries can be satisfied using any possible method.

2. Literature Survey

Real-time Query Scheduling for Wireless Sensor Networks By Bo Sheng and Qun Li (IEEE Transaction on Network Security, 2011

- Here, we show that there is an inherent trade-off between prioritization and throughput in conflict-free query scheduling.

Manuscript received June 5, 2014
Manuscript revised June 20, 2014

- The non-preemptive query scheduling algorithm achieves high throughput while introducing priority inversions.
- The preemptive query scheduling algorithm eliminates priority inversion at the cost of reduced throughput. The slack stealing query scheduling algorithm combines the benefits of preemptive and nonpreemptive scheduling by improving the throughput while meeting query deadlines.

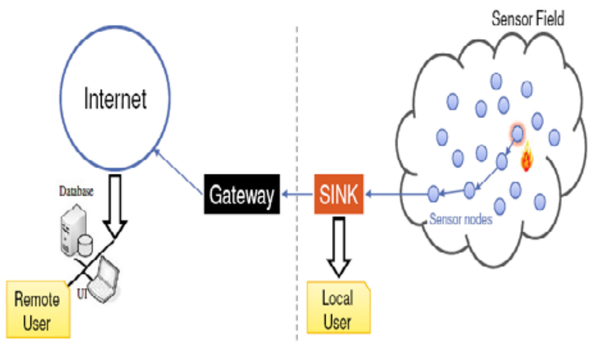


Figure 1: Architecture of wireless sensor network

3. Related work, our preliminaries

We will present our preliminary result, which serves as a basis for our protocol design.

A. Real-Time Scheduling

Two classes of well-studied real-time scheduling algorithms are rate-monotonic (RM) and EDF scheduling. RM assigns static-priorities to queries on the basis of the cycle duration of the jobs.Here we presented an 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) , or similar one [2], is to judiciously adjust the local deadlines of packets at a node, 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.

Next, we present our preliminary result of packet labeling for single-hop queries. Here, a *single-hop query* differs from the query in only one aspect: Each packet requests only a single-hop transmission (instead of multihop transmissions across the network). We define the *request rate* of a single-hop query as the reciprocal of its period. Let the utilization of a set of queries be the summation of their request rates.

Definition of Packet Labeling: Given a set Q'of preemptive and periodic single-hop queries, the objective is to assign a different integer label for each packet, such that if each packet transmits at the time-slot equal to its label, the delay for each query is at most its period.Observe that a packet labeling scheme corresponds to a single processor periodic job scheduling. Thus, we can label packets based on RM or EDF scheduling.

RM Scheduling: RM prioritizes packets simply based on request rates of queries.When the number of queries is large, RM scheduling can achieve a utilization of 69% (all packets can make their deadlines).

EDF Scheduling: EDF prioritizes packets strictly according to their deadlines. EDF can achieve utilization of exactly one [11].

B. Min-Delay Aggregation Scheduling

Minimum delay data aggregation problem has been proven to be NP-hard [4], even for the case of simple one-shot query. Authors proposed a sequence of constant-ratio approximation algorithms for OQS under PrIM. Based on the related work, we will present our preliminary result of node ranking in a connected dominating set (CDS) (see the definition of CDS in [18]).

In the wireless network community, several interference models have been commonly adopted, e.g., *Protocol Interference Model* (PrIM), *RTS/CTS Model*. In PrIM [7], each node has a fixed *transmission range* normalized to one and a fixed *interference range* of ρ . Any node v € V will be interfered by the signal from another node u € V if $\|uv\| \leq \rho$ and the node v is *not* the intended receiver of the transmission from u. In the RTS/CTS model [1], for every pair of active transmitter and receiver, any other node that lies within the interference range of either the transmitter or the receiver cannot transmit simultaneously.

Definition of Node Ranking: Given a CDS , sink , interference model, the objective is to assign a rank for each node in CDS, such that if all nodes transmit toward the sink at the time-slot equal to its rank, the aggregated datafrom the CDS can be received by without interferences. Clearly, given a CDS, a transmission schedule for OQS with the CDS as input graph corresponds exactly to a node ranking scheme. Moreover, the delay of the schedule corresponds to the maximum rank among all nodes in the CDS. We can compute the ranks of nodes based on existing solutions for OQS. We will focus on a CDS whose maximum node degree is bounded by a constant 12.

4. Scheduling protocol design

The general framework of our protocol design is universal for various wireless interference models. It consists of three phases.

1) For each query Qi € Q, construct a routing tree Ti for data aggregation.
2) For each node, construct a *transmission plan*, which specifies the data to transmit at the current moment. For each
query Qi, based on routing tree Ti , each node u in Ti(*u* may not be a source node) needs to add data for each period
to its plan.
3) For each node *u*, assign time to transmit for each packet from *u*'s transmission plan. The assignment will rely on
our preliminaries of Packet Labeling and Node Ranking . This phase is the key part.

The first phase is *routing*. For each aggregation query Qi € Q, the routing tree Ti should be a *Steiner Tree* interconnecting the terminals of Si U$\{v_s\}$. Given a communication graph, we first select a CDS of G.We then construct a *spanning tree* T_G by connecting each node *u* not in the CDS to one of *u*'s neighboring dominators. For each query Qi € Q , starting with T_G, we prune each node *u* € Vand an incident communication link uv (from *u* to its parent node *v*) in T_G, if the intersection of two node sets Si and the node set from the subtree of T_G rooted at *u* (noted as T_G^U) is empty: Si∩V($T_G^{U)}$ =Ø. The pruning operation results in a routing tree Ti for the query Qi.
The second phase is constructing *transmission plans*, based on routing trees for aggregation queries. For each query Qi € Q with a routing tree Ti , during each period, first each leaf node in Ti adds the source data to its transmission plan. Then, every internal node in Ti (noted as a *relay* node for query Qi) only generates one unit of data by aggregating all received data with its own data (if it has),while it may receive multiple data units from its children. Note that, before adds the data unit to its transmission plan, it needs to wait until receiving the corresponding data from all its children in (the routing tree for query). Thus, for a query , the data unit at node can be either: 1) original, or 2) an aggregated one, depending on whether this data unit comes from one node.
The third phase is *packet scheduling* at each node that contains data units in its transmission plan. We divide nodes into two complementary groups: nodes not in the CDS T_{CDS} (noted as *leaf nodes*) and nodes in T_{CDS} (noted as *intermediate nodes*). We will ensure that all leaf nodes transmit at even time-slots only, and all intermediate nodes transmit at odd time-slots only; the time-disjoint property can avoid interferences between nodes from different groups.

Packet Scheduling at Leaf Nodes: We employ a grid partition of the deployment plane. The vertical lines x = *i* .λ for *i* € Z and horizontal lines y = j. λ for j € Z partition the planes into half-open and half-closed grids of side λ (here, λ represents λ(μ), and Z represents the integer set) {[i. λ,(i+1) .λ X[j λ,(j+1) . λ : i , j € Z}
We then color the grids such that up to one node from every grid with a monotone color can transmit simultaneously. The number of colors used here (noted as c_2 (μ)) depends on the interference model μ .Under PrIM, RTS/CTS model, no neighboring grids sharing a common color is enough to avoid interferences,i.e., c_2 (μ) = 4 ; while under PhIM,c_2 (μ) is a larger constant (see [4]). We index the colors used and denote σ_g as the color of grid g(σ_g € {0,1,…, c_2 (μ) – 1}).
For every leaf node *u* , let σ_g be the color index of the grid g where *u* lies. We assign the *p*th packet for query
Q_i's th job at node *u* with the following transmission time: $t_u^{<qi,j,p>}$ = $l_u^{<qi,j,p}$>+2σ_g.This finishes packet scheduling at leaf nodes.

Packet Scheduling at Intermediate Nodes: For each intermediate node *u* , we find the set of queries Qu for which *u* participates in routing and map each query Qi € Qu to a new one Qi' with modification of only the release time:ai'=ai+Pi+2c_1c_2. Let Q_u ' = U $_{Qi € Qu}$ Q_u ' . We then create an instance of Packet Labeling with the single-hop query set Q_u '. Note that, for such an instance of Packet Labeling c_1 (μ) . ∑ $_{Qi €Q}$ χi / pi. we can obtain a packet labeling scheme by using RM (or EDF) scheduling where each packet (say the *p*th packet for query Q_i's jth job at the intermediate node *u*) is assigned with a label $l_u^{<qi,j,p}$> such that (2c_1) | $l_u^{<qi,j,p}$> (i.e., 2c_1 divides $l_u^{<qi,j,p}$>). For each intermediate node *u*, for each packet (assume it is the *p*th packet for Q_i's jth job), we assign a time-slot: $l_u^{<qi,j,p}$> +2r(u) + 1. Here,r(u) is the rank of *u* for Node Ranking with the CDS T_{CDS} as the input. This finishes packet scheduling at intermediate nodes.

TABLE I PARAMETERS USED IN OUR PROTOCOL DESIGN

T_i	routing tree for query Q_i
ρ	interference range for PrIM
$\mathcal{M}$	interference model *i.e.*,PrIM, CTS/RTS, PhIM
$\lambda(\mathcal{M})$	conflict range
σ_g	color index of the grid g
$\ell_u^{\langle \mathbf{q}_{i,j},p\rangle}$	label of p-th packet for Q_i's j-th job at node u in Packet Labeling
$r(u)$	rank of node u in Node Ranking
$\mathcal{U}_{G,\mathcal{Q}}(g_{v,h})$	initial load of grid $g_{v,h}$
$c_1(\mathcal{M})$ or c_1	refer to Lemma 2
$c_2(\mathcal{M})$ or c_2	number of colors for grid coloring
$c_3(\mathcal{M})$ or c_3	maximum # of nodes transmitting in a grid
$t_u^{\langle \mathbf{q}_{i,j},p\rangle}$	time assigned to p-th packet for Q_i's j-th job

To sum up, we present Algorithm 1 for our protocol design and Table I for the notations.

Algorithm 1: Scheduling Protocol for PAQS

Input: A set of periodic aggregation queries ,
an interference model μ.

for *each query Qi* **€** *Q* **do**
construct a data aggregation routing tree T_i;
for *each j th instance of each query Qi* **€** *Q* **do**
for *each node u* **do**
if *is a leaf node in* T_i **then**
adds the data to *c* 's transmission plan;
if *u is an internal node in* T_i **then**
if *u has only one child v in* T_i **then**
when *u* received the data from *v* ;
adds data to *u* 's transmission plan;
else
when *u* received data from all children
in T_i, generates aggregated data;
adds the data to *u*'s transmission plan;
for *each leaf node* (i.e., *u* € T_{CDS}) **do**
$\sigma_g \leftarrow$ color index of the grid g where *u* lies;
for *packet in u* (*p th packet for Qi's j* th *job*) **do**
assign time: $t_u^{<qi,j,p>} \leftarrow l_u^{<qi,j,p>} + 2\sigma_g$;
for *each intermediate node* (i.e. *u* € T_{CDS},) **do**
for *packet at* (p th *packet for Qi's j* th job) **do**
assign time: $t_u^{<qi,j,p>} \leftarrow l_u^{<qi,j,p>} + 2r(u) + 1$;
return *Time to transmit for each packet at each node.*

---

5. DISCUSSIONS

We have proved that our methods can achieve a total rate that is at least a constant factor of the optimum load. There are still a few limitations in our work, which will be summarized as follows. This will serve as some future research challenges.

First, we assumed that a node can aggregate any number of packets into a single packet, while in practice, a different *aggregation degree* may be used, i.e., the size of the aggregated data depends on the number of input data items. One challenge of extending the algorithm to different aggregation degree is to prove its performance.

Second, we omitted the extra *aggregation delay*. To address this practical challenge, one possible approach is sending a partially aggregated packet without waiting for data packets from all its children nodes. However, this approach may not improve the delay performance; it may hurt it actually. Note that the delay here is defined as the last time the sink collected the data.

In addition, there are some other challenges such as the following:1) the impact of unreliable network: During data transmissions, sensor nodes and links may suffer from packet losses, which will often trigger rerouting and retransmissions of data. This will incur additional delay and overhead to the network; 2) the impact of the time synchronization errors on the performance of the proposed methods.

Conclusion

Real-time queries appear in many sensor network applications. For answering periodic queries, we proposed a set of efficient scheduling schemes for data communications. Essentially, we jointly designed the routing strategy as well as packet scheduling protocols under various interference models. Most importantly, we theoretically proved that our algorithm can achieve constant approximation in terms of both delay and schedulability.

References

[1] G. Brar, D. Blough, and P. Santi, "Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks," in *Proc. ACM MobiCom*, 2006, pp. 2–13.

[2] X. Chen, X. Hu, and J. Zhu, "Minimum data aggregation time problem in wireless sensor networks," in *Proc. LNCS*, 2005, pp. 133–142.

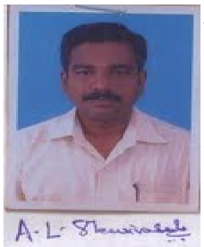
[3] O. Chipara, C. Lu, and G. Roman, "Real-time query scheduling for wireless sensor networks," in *Proc. IEEE RTSS*, 2007, pp. 389–399.

[4] X.-Y. Li,X.Xu, S.Wang,S.Tang, G. Dai, J. Zhao, andY. Qi, "Efficient data aggregation in multi-hop wireless sensor networks under physical interference model," in *Proc. IEEE MASS*, 2009, pp. 353–362.

[5] X. Xu, X.-Y. Li, P.-J. Wan, and S. Tang, "Efficient scheduling for periodic aggregation queries in multihop sensor networks," Illinois Institute of Technology, Chicago, IL, 2011 [Online]. Available: http:cs.iit. edu/~xli/paper/Journal/query-TON2011.pdf.

[6] P.-J.Wan, S.C.-H.Huang,L.Wang, Z.Wan, andX. Jia, "Minimum-latency aggregation scheduling in multihop wireless networks," in *Proc. ACM MobiHoc*, 2009, pp. 185–194.

B.Madhavilatha obtained computer science engineering degree (B.E) in 2010 from intell engineering college,JNTUA University, Anantapur , Andhrapradesh , India. Currently she is pursuing her M.E (Computers science and Engineering) in the same college, Anantapur , Andhrapradesh , India



A.L.Sreenivasulu pursuing Ph.D in computer science from JNTU Anatapur and got 14 years of teaching experience and guided several projects both UG & PG.His areas of interest include wireless sensor networks,computer networks and information security.