Poster: Real-time Message Scheduling with Multiple Sinks

Seong-eun Yoo
School of Computer and Communication Engineering,
Daegu University, Korea
seyoo@daegu.ac.kr

Cormac J. Sreenan
Department of Computer Science,
University College Cork, Ireland
cjs@cs.ucc.ie

Abstract

A smart factory in Industry 4.0 requires the exchange of sensing and control messages over a wireless sensor network. Communication typically has real-time constraints and the nodes are densely deployed. Thus, multiple sink nodes are necessary in order to ensure schedulability. This paper deals with the multiple sink scheduling problem, requiring an allocation policy for messages over the sinks, as well as per-sink message scheduling. The objective of the optimization is to jointly minimize the number of the required sinks while respecting the limitations of the underlying protocols and scheduling the real-time message set. We present a heuristic algorithm and preliminary results that demonstrate its efficacy.

1 Introduction

In an Industry 4.0 smart factory it is expected that sensing and control messages will be exchanged in networks comprised of a dense deployment of nodes. Most messages will have real-time delivery requirements. Using a single sink it is challenging to achieve schedulability [3]. Thus, multiple sinks are a natural remedy. This requires a solution to the allocation problem - how messages should be allocated across multiple sinks. It also raises the question of how to determine the required number of sinks. A naive approach for determining the number of sinks is to use a calculation based on the estimated data rate requirements and the capacity of a sink. For example, a network engineer might use $N_{ReqSk} = \frac{U_M}{C_{PSk}}$ to estimate the number of required sinks, where U_M is the total utilization of the message set and C_{PSk} is the capacity per sink. While straightforward, this considers neither the protocol overheads nor the deadline (or period, assuming the deadline of a message is the same as its period) and so cannot guarantee schedulablity. Separately, the allocation problem remains unanswered. Therefore, this paper addresses the multiple sink scheduling problem in a systematic way, ensuring a requisite number of sinks and that messages are allocated and scheduled to meet real-time delivery targets. To provide a grounding in the problem and solution, we turn our attention to the well researched topic of multiprocessor scheduling. These algorithms can be categorized into the partitioned approach and the global approach [1]. Unlike the partitioned approach, the global one permits the migration of a job or task, but the message scheduling in a network may incur too much overhead by allowing the migration of the instances of a message as the global approach. Therefore, this paper proposes a partitioned harmonic rate monotonic multiple sink scheduling algorithm, but considering the underlying protocol constraints, and generating a scheduling information for each sink. There is some related work, for example on superframe scheduling [2] and real-time message scheduling for a single sink in the industrial internet of things [3]. In contrast, this paper addresses the multiple sink scheduling problem and proposes a systematic solution for the aforementioned primary questions.

International Conference on Embedded Wireless Systems and Networks (EWSN) 2018
14–16 February, Madrid, Spain

ISBN: 978-0-9949886-2-1

2 Problem and Proposed Algorithm

In the current solution, we assume a beacon-enabled network and that every sensor node can reach any sink node with the equivalent link quality and that sinks operate on different channels and do not interfere. As an example of a beacon-enabled network, we adopts IEEE 802.15.4, which has a simple but useful feature to support a contention-free period (CFP) with superframe structure [3]. Two standard specific parameters, BO (Beacon Order) and SO (Superframe Order), define the superframe. BO specifies the BI (Beacon Interval), the interval of two consecutive beacons, while SO defines the active portion and the slot size. Each superframe can support up to 7 GTSs (Guaranteed Time Slots) in the CFP. Each message has two application-level attributes, the length (in bytes) and the period (or deadline, in seconds). The real-time message set (**M**) comprises all of the N messages to be scheduled.

The problem we solve is to minimize the number of required sinks to schedule **M** while respecting the real-time constraints of the set and of the underlying communication protocol. The inputs are **M**, with N number of messages, and the protocol constraints, e.g. IFS, *aMinCAPLength*, and up to 7 GTSs in a superframe. Schedulability is determined. In the schedulable case, the scheduling table for each sink node as well as the required number of sinks are identified. We design a heuristic Partitioned Harmonic Rate-monotonic (PHR) Scheduling Algorithm, see Algorithm 1.

PHR takes the message set **M** and ProtocolConstraints as input. In line 3, it sorts the message set, adds the protocol overheads (headers in MAC/PHY layers, etc) to the length of each message, and changes the attribute units of each message in symbol. Line 7 makes sure that the total required real-time demand of the message set is less than the capacity of the maximum available sinks. From line 9-18, the algorithm allocates and schedules the message sets using the available sinks. Depending on the allocation policies, i.e. Next-Fit (FF) or First-Fit (FF), as employed in bin packing algorithms, a different subprocedure is called. If a message is not schedulable in a sink, NF policy opens a new sink and tries to allocate the subsequent messages to the new sink, while the FF policy checks the scheduliability for the open sinks first. If the message is not schedulable, FF opens a new sink and tries allocating the subsequent messages. Before finishing this section, we explain HScheduleAsManyMsgAsPoNewSink more in detail. It first harmonizes (makes every larger period divisable by every smaller period) the message set with respect to BI [3] to guarantee 100% of the channel utilization and schedules each message to meet the real-time constraint and the protocol constraints.

Algorithm 1 Partitioned Harmonic RM algorithm

1: **procedure** PHRSCH(**M**, AlPol, ProtocolConstraints)
2: //using 0.5 byte/symbol and 16us/symbol @ 2.4GHz, IEEE 802.15.4
3: **MS** $\leftarrow$ SortNAddOverhead2Symbol(**M**) //to make it in rate-monotoic
4: $BO \leftarrow$ CalculateInitialBOBasedOn*min(PS)*
5: **if** $BO > 14$ **then return** fail(minBI) //min(PS) $<$ minBI
6: $SO \leftarrow BO$ //To use full duty cycle
7: **if** *UtilizationCheck()*! $=$ *PASS* **then return** fail(UCheck)
8: **OSink** $\leftarrow$ OpenNewSink()
9: **while** AnyUnscheduledMsg **do**
10: $result \leftarrow HScheduleAsManyMsgAsPoNewSink()$
11: **if** $result == SUCCESS$ **then return** success
12: **if** AlPol $== FF$ **then**
13: $result \leftarrow HScheduleAsManyMsgAsPoOpenSinks()$
14: **if** $result == SUCCESS$ **then return** success
15: **if** $|$**OSink**$| == MxASk$ **then**
16: **if** BO **==** 0 **then return** fail(Max)
17: **else** $BO \leftarrow BO$ - 1 and **OSink** $\leftarrow$ 0 and **goto** *STEP 6*
18: **else OSink** $\leftarrow$ **OSink** $+$ OpenNewSink()
19: **return** success

3 Preliminary Results

To study the efficacy of the PHR algorithm, we implemented a simulation and evaluate the performance in terms of the required sinks and the schedulability. The maximum number of sinks is set to 16, corresponding to the maximum number of available channels in IEEE 802.15.4. We generate random message sets based on the application level utilization (i.e., real-time demand). For example, to generate a message set **M** (N = 60) with the total real-time demand of 0.01, we generate 60 uniform random messages distributed from 1 byte (1B) to 102B and calculate the period of each message to distribute the total utilization of each message randomly. We generate 100 message sets for each demand and check the required number of sinks and the schedulability as a simulation run, and we execute an experiment with 50 simulation runs and show the average values. We experiment by increasing the real-time demand from 0.01 to 2.17 with a step of 0.06. Figure 1 shows the required number of sinks to schedule the message sets according to the different real-time demands for N = 30, 60, and 120. As we increase the real-time demand, the required number of sinks increases. Since it increases monotonically, we omit the results from 0.37-1.03. In a smaller utilization, the allocation policy does not show meaningful gain, for there aren't enough open sinks for FF. In a higher utilization, FF requires less sinks to schedule the message set. In a smaller N, the FF shows the higher gain since even a small number of messages scheduled in the open sinks reduces the required number of sinks more easily. Using the aforementioned naive estimation for the number of sinks required, a network engineer could easily underestimate the required number of sinks as the red-dotted line (U_C) in Figure 1 shows. Of course, it does not consider the real-time requirements. Figure 1 also shows the schedulability (i.e., the ratio of **M** meeting the deadlines) for the same experiments. As in the required numbers of the sinks, we have more gain in the schedulability with FF in a smaller N and see the higher schedulable utilization with FF.

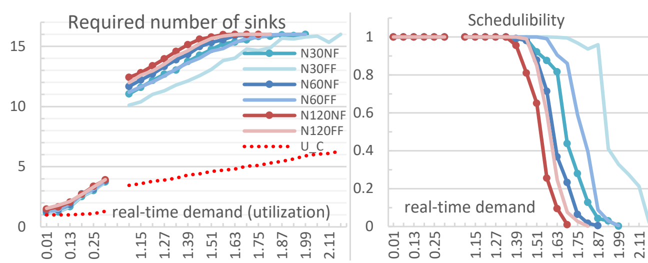

Figure 1. Required number of sinks and Schedulability

4 Conclusion

We address the joint problems of determining the minimum number of sinks and message allocation so as to achieve real-time message delivery in dense Industry 4.0 networks. Preliminary results demonstrate the efficacy of our Partitioned Harmonic Rate-monotonic algorithm. Future work will seek improvements by considering the slot usage efficiency and relaxing the assumptions in the problem statement so as to anticipate practical deployments.

5 Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2016R1D1A1B03930379) and by Science Foundation Ireland Grant No. 13/RC/2077

6 References

[1] R. I. Davis and A. Burns. A survey of hard real-time scheduling for multiprocessor systems. *ACM computing surveys*, 43(4):35, 2011.

[2] E. Toscano and L. L. Bello. Multichannel superframe scheduling for ieee 802.15. 4 industrial wireless sensor networks. *IEEE Transactions on Industrial Informatics*, 8(2):337–350, 2012.

[3] S. Yoo et al. Guaranteeing real-time services for industrial wireless sensor networks with ieee 802.15.4. *IEEE Transactions on Industrial Electronics*, 57(11):3868–3876, Nov 2010.