Symphony - Routing Aware Scheduling for DSME Networks

Harrison Kurunathan, Ricardo Severino, Anis Koubaa, Eduardo Tovar
CISTER/ISEP and Prince Sultan University
Porto, portugal and Saudi Arabia
{hhkur,rarss,emt}@isep.ipp.pt,akoubaa@psu.edu.sa

ABSTRACT

Deterministic Synchronous Multichannel Extension (DSME) is a prominent MAC behavior first introduced in IEEE 802.15.4e. It can avail deterministic and best effort Service using its multisuperframe structure. RPL is a routing protocol for wireless networks with low power consumption and generally susceptible to packet loss. These two standards were designed independently but with the common objective to satisfy the requirements of IoT devices in terms of limited energy, reliability and determinism. A combination of these two protocols can integrate *real-time QoS demanding* and *large-scale IoT networks*. In this paper, we propose a new multi-channel, multi-timeslot scheduling algorithm called Symphony that provides QoS efficient schedules in DSME networks. In this paper we provide analytical and simulation based delay analysis for our approach against some state of the art algorithms. In this work, we show that integrating routing with DSME can improve reliability by 40 % and by using Symphony, we can reduce the network delay by 10-20% against the state of the art algorithms.

KEYWORDS

IEEE 802.15.4e, DSME, Multisuperframe scheduling

1 INTRODUCTION

Modern embedded systems, coupled with the advancements of digital communication technologies, have been enabling a new generation of systems, tightly interacting with the physical environment via sensing and actuating actions: Cyber Physical Systems (CPS). These systems, characterized by an unprecedented levels of ubiquity, have been increasingly relying upon wireless communication technologies to provide seamless services via flexible cooperation, supporting different *Internet of Things* (IoT) applications. Several of these applications demand increased Quality of Service (QoS), namely regarding determinism, reliability, scalability and no compromise on energy efficiency.

The IEEE 802.15.4e standard provides time critical support for IoT applications by introducing new MAC behaviors like TSCH, DSME and LLDN [10]. Among these MAC behaviors, ***DSME*** - *Deterministic Synchronous Mutichannel Extension* is a very versatile MAC behavior. Like the classic IEEE 802.15.4, it can alternate between CSMA/CA and Guaranteed Timeslots (GTS) to support both best effort and time-critical communications. DSME introduces several features like the multichannel access to increase the scalability and robustness of the network manifold. Despite its many enhanced features, the standard does not specify any network layer for QoS centric routing purposes. Although it can support mesh topology, no intuition is given regarding the right mechanism that can dynamically setup the necessary service.

Integrating a distributed routing protocol like RPL over DSME "helps achieving increased scalability (via routing), while providing robustness to cope with network changes". The challenge lies in the integration of these standards and providing DSME schedules periodically. In this paper we present an approach to integrate DSME with RPL and an algorithm called Symphony to deliver schedules periodically for the DSME associated nodes.

The main contribution in this paper are as follows:

- We overview the DSME and RPL networks and provide a system architecture for efficient integration of these standards.
- We introduce *Symphony*, a time-frequency algorithm that helps DSME nodes to maintain schedules periodically with dynamic changes in the network based on RPL.
- Using simulations we show the advantage of RPL over a traditional DSME network in terms of reliability.
- We use simulations to learn the advantages of Symphony over the state of the art algorithms in terms of delay.

The rest of the paper is structured as follows: in section II, we provide a brief literature survey then in Section III, we give an overview of DSME and RPL, then in Section IV we elaborate the system architecture of RPL over DSME. In Section V, we introduce and discuss our algorithm Symphony. Finally, we provide an in-depth performance analysis of our architecture and compare Symphony with some of the state of the art algorithms for DSME scheduling.

2 RELATED WORKS

Following the standardizing efforts on protocols like 6LoWPAN [12], the *Internet Engineering Task Force* (IETF) has focused on implementing 6TiScH [4], a combination of the TSCH MAC behavior of IEEE 802.15.4e, IPV6 and RPL. Implementing RPL over these standards helped in providing optimal routing for the transmissions and increased the overall reliability. *Orchestra* [5] is one of the open source implementations based on 6TiScH, in which, the nodes automatically compute their own local schedules and maintain several schedules for different traffic scenarios. Orchestra was able to deliver high end-end delivery ratios with a good latency-energy balance. In our work we provide an architecture for the implementation of RPL over DSME networks.

The DSME MAC behavior of IEEE 802.15.4e provides increased determinism and reliability in a multi channel environment. Several researchers like in [13] and [15] have demonstrated the advantages of DSME in terms of lesser delays and aggregate throughputs compared with standard IEEE 802.15.4.

RTN'19, July 2019, Stuttgart, Germany

There is some literature on developing scheduling algorithms for the enhancements of IEEE 8021.5.4e to provide an optimal service. For example, in case of TSCH, an other prominent MAC behavior of IEEE 802.15.4e, a new enhancement called *Adaptive-TSCH* [3] was developed by Peng Du. In this algorithm, the author provides the nodes, the ability to hop amongst a subset of channels which are deemed reliable based on their respective link qualities. Using this technique an average increase of ETX (Expected Transmission Count) by 5.6 % was observed.

There is also some research in implementing multi channel scheduling algorithms for DSME [14] to improve its reliability. In this algorithm several dummy GTSs slots were allocated to occupy the transmissions in case of a transmission failure. However, this approach can impact over the overall delay of the network. In this paper we compare this scheduling algorithm with Symphony.

Several researchers [9], [1] in their work developed analytical and simulation assessments of DSME and TSCH MAC behaviors. They proved that DSME performs better than TSCH in terms of end to end latency when the number of nodes is higher than 30. The enhanced features of DSME like CAP reduction helped in reducing the end to end latency and also achieving better throughput and scalability.

In this paper, we propose merging the functionalities of DSME and RPL and aim at reducing the latency of the overall network. RPL will provide optimal routes based on any objective function such as power efficiency or link reliability, while our proposed algorithm *Symphony* will provides dynamic GTS schedules periodically for the entire network with minimal delay.

3 BACKGROUND TO DSME AND RPL

The DSME network provides deterministic communication using its beacon enabled mode. This beacon enabled mode is supported by multisuperframes that comprises stacks of superframes as shown in Figure 1. Every superframe comprises of a Contention Access Period (CAP) in which the nodes contend to access the channel and a Contention Free Period (CFP) in which the nodes send the data using Guaranteed timeslots (GTSs).

The superframe is defined by *BO*, the *Beacon Order* which is the transmission interval of a beacon in a superframe, *MO* the *Multi superframe Order* that represents the enhanced beacon interval of a multi-superframe and *SO* the *Superframe Order* that represents the beacon interval of a superframe. The number of superframes in a multisuperframe can be given by $2^{(MO-SO)}$. The values of BO, SO and MO are set by the PAN coordinator and is conveyed to the nodes via an Enhanced Beacon (EB) at the beginning of each Multisuperframe. This EB helps in the overall synchronization of the network.

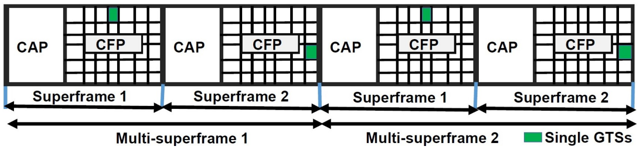


Figure 1: Superframe structure with *BO*=3, *MO*=3, *SO*=2

DSME can use *channel adaptation* or *channel hopping* for multichannel access in the CFP. In channel hopping, the hopping-sequence of the channels for data transmission is pre-determined and the same hopping pattern is repeated till the end of the data transmission. Whereas in channel adaptation, the transmissions are allowed to hop over the channels based on their link quality. The multichannel access mechanisms of DSME allow several transmissions to occur in the same timeslot within different channels. These multichannel access schemes open the possibility of forming complex topologies like mesh for DSME networks.

RPL is a routing protocol that integrates technologies like IEEE 802.15.4 and IPv6 protocols. It supports both mesh as well as hierarchical topologies, and is specifically designed to support networks that are prone to high exposed packet losses and limited resources in terms of computation and energy.

RPL is based on hierarchical Directed Acylic graphs (DAGs) in which a node can associate itself with many parent nodes. The destination node of an RPL is called a sink and the nodes through which a route is provided to internet are called gateways. RPL organizes these nodes as Destination-Oriented DAGs (DODAGs). In an RPL, every router in the system identifies and associates with a parent. This association is done based on an Objective Function (OF). *OF* can be based on quality determining parameters like LQI (Link Quality Indicator) and RSSI (Received Signal Strength Indicator). OF helps in providing an optimal routing path using metrics like latency or power efficiency.

4 SYSTEM MODEL

In this paper, we introduce *Symphony* a dynamic algorithm that provides *"a variety of schedules to fit onto the multichannel DSME - GTSs based on optimal routing decisions made by RPL."*

RPL can use either broadcast or unicast to disseminate the Objective Function metrics using the DODAG Information Object "*DIO*". This information also can be requested using the DODAG Information Solicitation "*DIS*". The routing paths can be disseminated using a Destination Advertisement Object "*DAO*". In an RPL network perspective, when a node wants to join the DODAG it receives a signaling message from a neighbor router, it (*i.*) adds the sender address to its parent list, (*ii.*) computes a rank according to the Objective Function such as reliability determining factors like LQI (Link Quality Indicator) or RSSI, (*iii.*) forwards the updated rank information to the parent.

For the system model we consider a mesh network (Figure 3) with fully functional devices (FFDs) that can receive and transmit messages in the Guaranteed Timeslots (GTSs). The FFDs maintain the schedules locally and have their own superframes to accommodate the nodes associated to them. They also have a routing table to maintain the nodes associated to them. Every superframe carries various kinds of traffic to support symphony, such as the periodic beacons for synchronization, RPL signaling traffic and application data traffic.

In case of a GTS allocation, the allocation-request is sent to the parent node (FFD) through the RPL network. The Symphony algorithm at the coordinators helps to find the most efficient allocation in the time-frequency domain. Symphony aims at "maintaining schedules for all the transmissions in parallel without a overlap". It chooses specific channels and timeslots for the GTSs transmissions in order to achieve a "interference and a contention free scheduling".

A concrete example of our architecture (Figure 2) is as follows:

- A dedicated beacon broadcast for synchronization between every superframe for every "X" slots, where "X" is the superframe duration of every individual superframe.
- A dedicated beacon broadcast for synchronization every multi superframe for every "Y" slots, where "Y" is the multi superframe duration coordinating every superframe with the duration of "X".
- A Enhanced Beacon common for all coordinators in the network carrying the broadcast + unicast packets for RPL signaling (DIO, DIS, DAO), repeating every "Y" slots. In accordance with the standard, the Enhanced Beacon payload can be a variable and it carries the RPL information.
- Dedicated unicast signal from the slave node to the parent node followed by N unicast signals from the coordinator to the slave nodes.

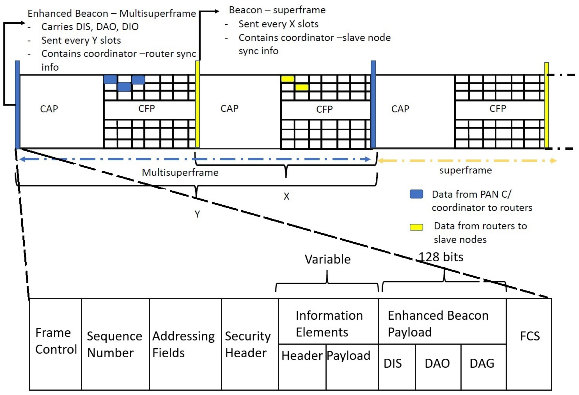


Figure 2: System Architecture

5 SYMPHONY ALGORITHM

Symphony is a routing aware algorithm that was designed based on the methods of solving a *Constraint Satisfaction Problem* (CSP). It performs scheduling based on several decision parameters like that of the classic *eight queens problem* [6]. The optimal assignment of time-slots and frequencies which is done by Symphony is considered to be an NP-Hard problem [7]. Symphony will aim at providing *"dynamic allocation of timeslots based on the routing information provided by RPL."*

This scheduling problem is bounded by two major constraints, which will be a determining factor in establishing an optimal solution.

Constraint 1: *No same nodes either involving in transmission or reception must fall under the same timeslot.*

This constraint helps in avoiding all the interference in the network. The standard offers a possibility for *different* nodes to communicate in a *same* timeslot simultaneously in *different* channels, whereas, the same nodes can communicate in *different* timeslots within the *same* or *different* channels.

Constraint 2 *Maximum number of channels and minimum number of timeslots should be used.*

This constraint is more of a "quality constraint" that helps in establishing the optimality of the algorithm. This constraint helps in achieving the fact that "more bandwidth will not be wasted" and at the same time "minimal timeslots will be used". By satisfying this constraint the overall network throughput and scalability of the network can be significantly increased, concomitantly achieving minimal latency.

For our analysis we take a mesh network with 5 different nodes that are interconnected with each other as shown in Fig 3. This topology is considered to be obtained through RPL. This network model can also be extended to any number of slave nodes with reduced functionality (only receive information). For the schedule placement, we only consider the guaranteed timeslots in the CFP region of the DSME superframe with 3 channels in our model.

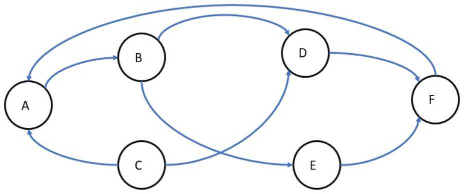


Figure 3: example of a mesh network

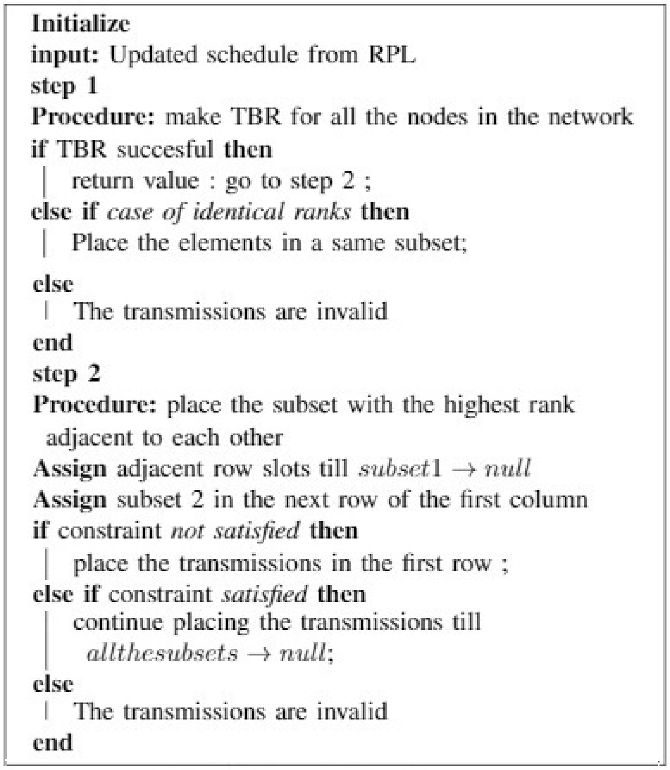
A schedule is considered to be optimal when it uses the resources stringently and fully utilizing the multichannel capability of DSME (Constraint 2). The optimality is checked by the following equation:

$$NT = \lceil (n/C) \rceil \tag{1}$$

In the above equation NT represents the number of timeslots occupied, n represents the total number of transmissions and C is the number of channels used. It should also be noted that proving the optimality should satisfy both **Constraint 1** and **Constraint 2**. This optimal schedule can be obtained by an ILP formulation provided in [11].

Our algorithm is a two step process, first we get Transmission Based Ranks (TBR) for the nodes based on the number of routes determined by the RPL. For example, in Figure 3, nodes B and C have a transmission rank of 2, as both the nodes have two links formed from them. We denote this Transmission Based Ranking as *TBR* in our algorithm. As an output of TBR, we group several sets of transmissions based on their respective ranks. In case of identical ranks, we place the elements under a single subset. This step is done in order to avoid any interference conflict in the scheduling (Constraint 1). The subsets are grouped for all the transmission routes provided by the RPL. The algorithm can be used for any number of nodes that are associated with a PAN Coordinator or a router to form respective schedules for the superframe.

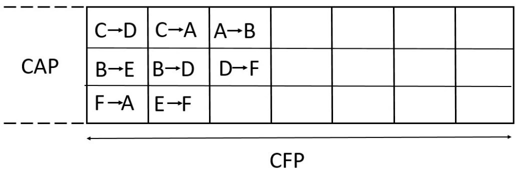
For the example provided in Figure 3, we start placing transmissions from C in adjacent timeslots (highest rank). By placing these elements in the adjacent timeslots, we can negate any chances of interference that can occur by the transmissions trying to transmit



```
Initialize
input: Updated schedule from RPL
step 1
Procedure: make TBR for all the nodes in the network
if TBR succesful then
    return value : go to step 2 ;
else if case of identical ranks then
    Place the elements in a same subset;
else
    The transmissions are invalid
end
step 2
Procedure: place the subset with the highest rank
 adjacent to each other
Assign adjacent row slots till $subset1 \to null$
Assign subset 2 in the next row of the first column
if constraint not satisfied then
    place the transmissions in the first row ;
else if constraint satisfied then
    continue placing the transmissions till
     $allthesubsets \to null$;
else
    The transmissions are invalid
end
```

Algorithm 1: Symphony

along the same timeslot. Further as the highest rank is placed initially, we devise a better strategy to accommodate the rest of the nodes in a more optimal way, so that less number of timeslots are utilized in scheduling. This step is now followed by the scheduling transmissions from B in the next channel of the same timeslot. This process is then backtracked to assign all the transmissions. Using this algorithm, we receive an optimal solution as shown in Figure 4



C→D	C→A	A→B				
B→E	B→D	D→F				
F→A	E→F					

Figure 4: Symphony schedule solution

6 PERFORMANCE ANALYSIS

Our performance analysis of this work is two fold: first we demonstrate the improvement in reliability with routing implemented over a DSME network. Then we use probabilistic analysis to calculate the delay and compare the advantages of symphony over several state of the art algorithms.

Every node in the network derives a ETX (Expected Transmission Count). This is a parameter that is helpful in estimating the frame loss ratio at the link. The ETX is dependent on the forward (P_f) and the backward frame losses (P_b) of the nodes in a network, and this value can be given by:

$$ETX = 1/(1 - P_f)(1 - P_b) \tag{2}$$

ETX can determine the reliability of the links as the parameter represents the inverse of successful packet delivery(P_S):

$$ETX = 1/(P_{Sf} \times P_{Sb}) = 1/Reliability \tag{3}$$

In an RPL enabled network, the nodes will change the routes to the sink when there is a deterioration of the link quality and eventually the overall ETX. The delay also can increase when more additional routes are deployed to reach the sink in case of a failure.

Using OpenDSME [8] an Omnet based simulation platform, we simulated the reliability over a network of 25 nodes with static concentric mobility type. Reliability of the network was calculated based on the number of successful packet delivery as shown in Equation 3. In the radio medium, we introduce a constant interference range to emulate a real-time wireless network. We used a payload of 75 bytes carried in 100 packets over 16 channels of the DSME network in accordance to the standard parameters. Without having routing established for the network layer, it was noted that the reliability of the network depletes steadily with the increase in the number of nodes. We repeated the same experiment with the same network configuration but with generic routing employed in the network layer. We were able to observe that the reliability does not deplete steadily and almost shows 40% betterment results (Figure 5).

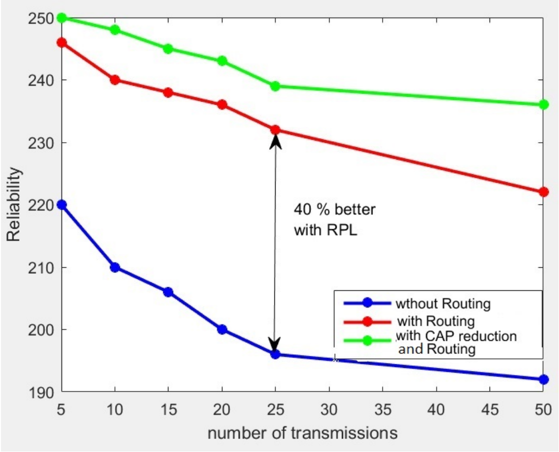


Figure 5: Reliability with generic routing

For the performance analysis of Symphony, we decided to caryout a probability based delay analysis and then complement our findings with simulations carried out in OpenDSME [8]. In both our numerical and simulation analysis we compared the performance of Symphony against state of the art algorithms like MDT [14], best effort DSME scheduling and Random FIFO.

The average transmission delay can be calculated for successfully transmitted GTS frames in the multisuperframe can be given by:

$$\delta = \sum_{i=0}^{\infty} P^{f}_{(i,m)}(i(MI)) \tag{4}$$

Considering the schedule for routing is carried our every multisuperframe, $P^{f}_{(i,m)}$ is the probability that the GTS is successfully transmitted in the i^{th} superframe of the multisuperframe m. MI is the summation of all the individual BIs (Beacon Intervals) within the multisuperframe. To calculate this probability let us take two parameters: X^{s}, the total number of GTS that is successfully transmitted, and $X^{S}_{(i,c)}$, the number of GTS that have to wait i superframes with c channels within a multisuperframe for its successful transmission. Using these parameters the probability $P^{f}_{(i,m)}$ can be formulated as:

$$P^{f}_{(i,m)} = \sum_{0}^{i} X^{S}_{(i,c)}/X^{S} \tag{5}$$

This probability considers the success of all the transmissions within the multisuperframe m. Considering that the first set of GTS frames based on the symphony schedule that gets successfully placed in the initial attempt, they need not wait another superframe interval for their data transmission. Let us consider this as $X^{S}_{(0,c)}$. The value of H varies depends on the success of this transmission.

$$\begin{gathered} X^{S}_{(0,c)} = H(1 - P_e), \\ where\ c = (0 - 16)\ and\ H\ \epsilon\ (0,1) \end{gathered} \tag{6}$$

The value of $X_{(i)}$ will be incrementing as with the failures to accommodate a successful transmission. The GTS superframes that wait till the first adjacent superframe to get transmitted successfully can be denoted by $X^{S}_{(1,c)}$, this value can be formulated as:

$$X^{S}_{(1,c)} = H(1 - P_e) \tag{7}$$

where, H is the probability of failure to get accommodated within the initial transmission.The value of H can be given as $P_e e^{-BI \cdot c \cdot i\lambda}$, this probability is with an assumption that all the transmissions shall be carried out within the multisuperframe with i superframes and c channels with a GTS arrival rate of λ. Generalizing for all the i superframes, the successful transmissions can be denoted as:

$$X^{S}_{(i,c)} = H^{(i)}(1 - P_e) \tag{8}$$

The value of the successfully transmitted GTS in a single superframe can be given as:

$$X^{S} = \sum_{i=0}^{m} H^{(i)}(1 - P_e) \tag{9}$$

using the aforementioned equations, the probability to be transmitted in the i^{th} superframe can be calculated as:

$$P^{f}_{(i,m)} = (1 - H) \cdot\ H^{i} \tag{10}$$

and the overall average delay of the network can be given as:

$$\delta = \sum_{i=0}^{m} (1 - H) \cdot\ H^{i}(i(MI)) \tag{11}$$

For the numerical analysis we consider a multisuperframe with 2 superframes over 3 channels. We also consider three arrival rates for the delay analysis. The increase in delay can be due to lesser arrival rates. Lesser arrival rates also can have a negative impact on the throughput of the network. However the multichannel feature in DSME contributes to lesser delay and larger throughput.

We now use the probabilistic approach to calculate the delay of schedule placement within a superframe. Unlike the calculation for the entire multisuperframe, this calculation must be carried out for every timeslot (T_s) of a single superframe. For this case, we take the value of H and replace with H_{tslot} which is the probability of failure to accommodate within the initial timeslot. This aforementioned value can be expressed as:

$$H_{tslot} = P_e e^{-T_s \cdot c \cdot i\lambda} \tag{12}$$

In order to generalize the aforementioned equation, let us consider that all the timeslots have an equal size for all the i superframes in the multisuperframe. Hence we can derive a formulation for the delay for single GTS that fails to occupy the first timeslot and moves to the next. Now we derive the delay for a timeslot to be:

$$\delta_{timeslot} = P_e e^{-T_s \cdot c \cdot i\lambda}(T_l)/(1 - P_e e^{-T_s \cdot c \cdot i\lambda}) \tag{13}$$

For numerical analysis, we compared symphony with MDT [14] and brute-force FIFO algorithms [2]. This method is also used for the GTS scheduling allocation in the OpenDSME framework [8] for DSME implementation. The analysis shown in Figure 6 provides the Transmission delay of the GTS frames for a set of transmissions for different arrival rates (25, 50, 100 Kbps). With the change in the topology of the network (addition of nodes), RPL updates a new set of transmissions to be scheduled in the following multisuperframe.

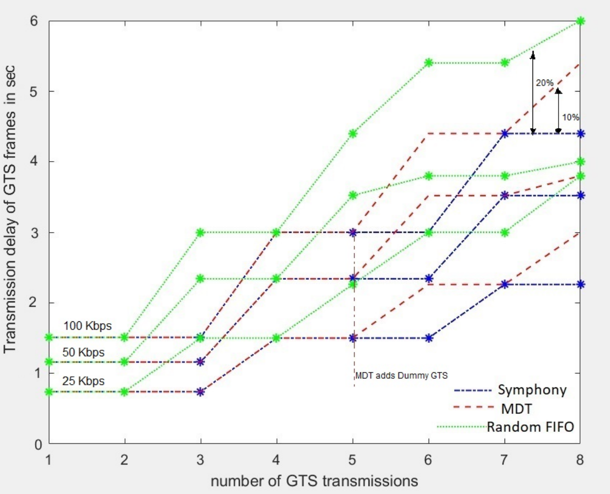


Figure 6: number of transmissions vs GTS delay (Analytical)

MDT under-performs because it spares timeslots aiming better reliability of the network. Certain amount of dummy timeslots are allocated for reliability purposes, contributing to the delay. These dummy packets result in more wasted bandwidth, eventually contributing to delay. The *Random FIFO* technique works based on best effort. In case of any conflict, the transmission is scheduled the eventual superframe to send the data. *Symphony* fills all the timeslots stringently on the basis of channels available, thus eventually leading to lesser transmission delays and also increased robustness. Unlike Random FIFO and MDT, the Symphony schedules did not wait until another Multisuperframe timeperiod to accommodate its transmissions. Hence, Symphony was stringently able to achieve lesser delay comparatively.

To complement our analytical results, we carried out simulations for Symphony using the OpenDSME platform. We conducted experiments for delay over several GTS transmissions. We simulated our experiments at a 100 Kbps traffic rate for varying number of transmissions. In our simulations, we pitted Symphony against MDT, standard DSME and CSMA/CA.

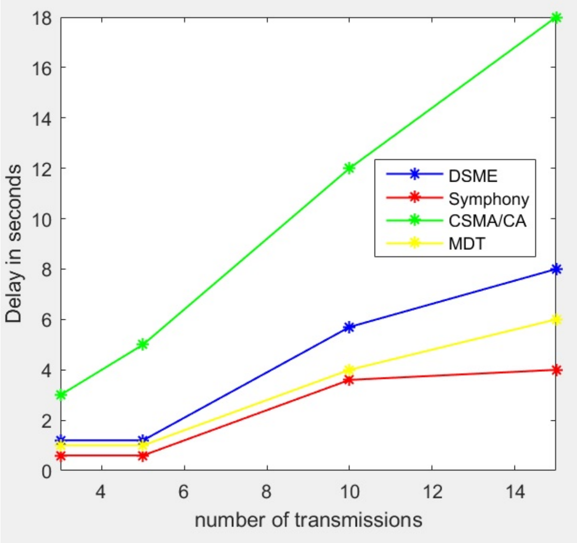


Figure 7: number of transmissions vs GTS delay

From our performance analysis and simulations, we learn that Symphony is able to achieve 10-15 % reduction delay when compared to many state of the art algorithms for DSME. As the number of transmissions increase Symphony is able to provide a schedule in such a way it is optimal to achieve a lesser latency. It also must be noticed that the transmissions that are provided onto Symphony is derived through RPL, which in-turn can improve the overall Quality of Service of the network manifold. We believe that integrating RPL onto DSME and providing a routing aware algorithm like Symphony can push DSME to become a de-facto standard for seamless IoT communication.

7 FUTURE SCOPE

In this paper we introduce an approach to improve the overall Quality of Service in a periodically evolving real-time DSME network. We provide an architecture for the integration of RPL and DSME technologies through a routing-aware algorithm called Symphony. The key goal of this work is to provide dynamic optimal schedules for GTS allocation based upon the RPL topology information, while reducing the latency of the overall network.

Through our detailed mathematical and simulation analysis we compared Symphony to some of the state of the art algorithms to find that, Symphony with its stringent packing strategy, performs better in terms of latency. By adopting symphony, we can witness a a decrease in latency by 10-15 %. Our Simulation of RPL also provides us an insight that routing over a dynamically evolving DSME networks can improve its reliability manifold.

We aim at implementing our algorithm in a hardware platform which will enable us to compare with the existing analytical results. We also intend to develop an open-source implementation of this protocol for Commercially Off The Shelf WSN platforms (COTS) (e.g. TelosB devices), to validate the results over real WSN hardware.

REFERENCES

[1] Giuliana Alderisi, Gaetano Patti, Orazio Mirabella, and Lucia Lo Bello. 2015. Simulative assessments of the ieee 802.15. 4e dsme and tsch in realistic process automation scenarios. In *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on.* IEEE, 948–955.

[2] Marc Domingo-Prieto, Tengfei Chang, Xavier Vilajosana, and Thomas Watteyne. 2016. Distributed pid-based scheduling for 6tisch networks. *IEEE Communications Letters* 20, 5 (2016), 1006–1009.

[3] Peng Du and George Roussos. 2012. Adaptive time slotted channel hopping for wireless sensor networks. In *Computer Science and Electronic Engineering Conference (CEEC), 2012 4th.* IEEE, 29–34.

[4] Diego Dujovne, Thomas Watteyne, Xavier Vilajosana, and Pascal Thubert. 2014. 6TiSCH: deterministic IP-enabled industrial internet (of things). *IEEE Communications Magazine* 52, 12 (2014), 36–41.

[5] Simon Duquennoy, Beshr Al Nahas, Olaf Landsiedel, and Thomas Watteyne. 2015. Orchestra: Robust mesh networks through autonomously scheduled tsch. In *Proceedings of the 13th ACM conference on embedded networked sensor systems.* ACM, 337–350.

[6] Robert W Floyd. 1967. Nondeterministic algorithms. *Journal of the ACM (JACM)* 14, 4 (1967), 636–644.

[7] Ted Herman and Sébastien Tixeuil. 2004. A distributed TDMA slot assignment algorithm for wireless sensor networks. In *Algosensors*, Vol. 3121. Springer, 45–58.

[8] Florian Kauer, Maximilian Köstler, Tobias Lübkert, and Volker Turau. 2017. OpenDSME-A portable framework for reliable wireless sensor and actuator networks. In *Networked Systems (NetSys), 2017 International Conference on.* IEEE, 1–2.

[9] Harrison Kurunathan, Ricardo Severino, Anis Koubâa, and Eduardo Tovar. 2017. Worst-case bound analysis for the time-critical MAC behaviors of IEEE 802.15. 4e. In *Factory Communication Systems (WFCS), 2017 IEEE 13th International Workshop on.* IEEE, 1–9.

[10] Harrison Kurunathan, Ricardo Severino, Anis Koubaa, and Eduardo Tovar. 2018. IEEE 802.15. 4e in a Nutshell: Survey and Performance Evaluation. *IEEE Communications Surveys & Tutorials* (2018).

[11] Harrison Kurunathan, Ricardo Severino, Anis Koubâa, Eduardo Tovar, et al. 2018. RPL over DSME: A Technical Report. CISTER.

[12] Nandakishore Kushalnagar, Gabriel Montenegro, and Christian Schumacher. 2007. *IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals.* Technical Report.

[13] Junhee Lee and Wun-Cheol Jeong. 2012. Performance analysis of ieee 802.15. 4e dsme mac protocol under wlan interference. In *ICT Convergence (ICTC), 2012 International Conference on.* IEEE, 741–746.

[14] Junhee Lee, Wun-Cheol Jeong, and Byeong-Cheol Choi. 2016. A multi-channel timeslot scheduling algorithm for link recovery in wireless multi-hop sensor networks. In *Information and Communication Technology Convergence (ICTC), 2016 International Conference on.* IEEE, 871–876.

[15] Tuomas Paso, Jussi Haapola, and Jari Iinatti. 2013. Feasibility study of ieee 802.15. 4e dsme utilizing ir-uwb and s-aloha. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on.* IEEE, 1863–1867.

[16] Prasan Kumar Sahoo, Sudhir Ranjan Pattanaik, and Shih-Lin Wu. 2017. A Reliable Data Transmission Model for IEEE 802.15. 4e Enabled Wireless Sensor Network under WiFi Interference. *Sensors* 17, 6 (2017), 1320.

[17] Thomas Watteyne, Antonella Molinaro, Maria Grazia Richichi, and Mischa Dohler. 2011. From manet to ietf roll standardization: A paradigm shift in wsn routing protocols. *IEEE Communications Surveys & Tutorials* 13, 4 (2011), 688–707.