QoS Assessment for Mission-critical Wireless Sensor Network Applications

Felix Dobslaw, Tingting Zhang, Mikael Gidlund
Mid Sweden University, Sweden
Email: {felix.dobslaw, tingting.zhang, mikael.gidlund}@miun.se

***Abstract*—Wireless sensor networks (WSN) must ensure worst-case end-to-end delay and reliability guarantees for mission-critical applications. TDMA-based scheduling offers delay guarantees, thus it is used in industrial monitoring and automation. We propose to *evolve* pairs of TDMA schedule and routing-tree in a cross-layer in order to fulfill multiple conflicting QoS requirements, exemplified by latency and reliability. The genetic algorithm we utilize can be used as an analytical tool for both the feasibility and expected QoS in production. Near-optimal cross-layer solutions are found within seconds and can be directly enforced into the network.**

I. Introduction

Wireless Sensor Networks (WSN) have the potential to diminish installation and maintenance costs for communication systems in the automation industry. WSNs increase mobility, flexibility and scalability as compared to the typically applied wired field-bus. However, for process automation, *true* wireless field devices are not foreseen in the near future, due to harsh requirements in noisy environments [1]. One main reason is that the WSNs available today were created for monitoring, not mission-critical applications. The challenge is to *guarantee* QoS via an unreliable communication medium, not energy efficiency, as power supplies can usually be expected near sensors. Mission-critical scenarios require updates every 10-250ms with a guaranteed reception failure below 0.001% [1]. The authors in [2] report a lack of analytical tools that display what is possible on the protocol level and suggest topological changes that satisfy the requirements. Cross-layer design strategies spanning routing and the MAC layer (even the physical layer) integrate the decision making of complementing means to efficiently realize QoS [3]. Here, the choice of the MAC layer protocol is critical, as it manages the access to the shared communication medium. ZigBee uses CSMA/CA on the MAC layer and has been shown to not perform well for the outlined applications, for instance, when compared to WirelessHART [4]. WirelessHART utilizes time-division multiple access (TDMA) on the MAC layer to guarantee the access to the communication medium [5]. Our contribution is a heuristic that rapidly finds near-optimal routing and TDMA cross-layer solutions considering multiple conflicting objectives. We simulate networks based on offline collected node-to-node connectivity statistics. The proposed genetic algorithm (GA) rapidly produces near-optimal lower-bounds, depicting the QoS that can be expected for the topology. In the time domain, we guarantee near-optimal end-to-end delay primal bounds for packet delivery in multi-hop networks in relation to finding as short a collision-free TDMA schedule as possible. Finding optimal solutions for this so called *scheduling problem* is NP-hard, even ignoring reliability and given a routing tree [6]. Reliability is addressed by end-to-end packet reception rates.

II. Related Work

Variants of the scheduling problem have been investigated previously (e.g. [6], [7]). For mission-critical applications, however, the problem is rather a constraint satisfaction problem involving reliability rather than a pure optimization problem. In [2], the authors present and compare recent QoS approaches for mission-critical WSN involving the MAC layer. Only a few contributions address the trade-off between latency and reliability. Even though WirelessHART supports QoS, the specification leaves it up to the vendor to select the appropriate routing and scheduling strategies for the central network manager. Further, the WirelessHART has no cross-layer design support; it *enables* collision-free communication on the MAC layer and reliability is improved by channel-hopping with no guarantees. The authors in [8] and [9] consider offline dimensioning including the a-priori assessment of noise levels. Neither tackle the problem in a cross-layer, but both use TDMA-based scheduling, considering burst-error metrics in order to improve the guarantee of latency-bounds by extending schedules with repeated slots. The greedy algorithm presented in [9] utilizes spatial-reuse to provide latency guarantees. Node-to-node packet reception rates (PRR_{n2n}) are not reported, which enables the assumption to be made that they were high. In environments with heavy machinery and noise, connectivity, on the contrary, can be poor at times, even if it is possible to obtain lower connectivity bounds. The authors in [10] present best-effort contention-based scheduling for multi-hop networks, using PRR_{n2n} and probability theory to maximize QoS for individual packets *online* by requesting *speed options*. A tool that, at deployment, suggests routing and scheduling combinations with performance guarantees for a user-defined confidence level, would be beneficial. None of the methods can provide such a-priori network guarantees.

III. Problem Formulation

We identify near-optimal TDMA schedule and routing-tree combinations by maximizing reliability and minimizing latency. By specifying user requirements, we will show how priority can be placed on either reliability, latency, or both (best-effort). Reliability is measured in end-to-end packet reception rates PRR_{e2e}, the ratio of packets that are sent at the beginning of a time-frame, and received at the sink at its end. Latency is measured as the length of the TDMA frame in slots. We assume a static single channel WSN with n sensors and a converge-cast scenario with a single data collector (sink). The presented cross-layer approach optimizes routing and scheduling simultaneously, and in making a decision as to extend schedules allowing for re-transmission at any node, or to re-route, if this is more appropriate. The physical layer is assumed to provide time synchronization and knowledge about node-to-node connectivity between any two sensors (PRR_{n2n}). A means of obtaining PRR_{n2n} is performed empirically, by collecting network statistics once the sensors are deployed (see e.g., [8], [9]). In addition, symmetric transmission and interference ranges are assumed, and in conformity with WirelessHART, transmissions and their ACKs fit into a single dedicated time-slot. The *up-link quality* Q_{PRR}=q_{ij} is defined as a matrix with n rows and n+1 columns containing, in position q_{ij} the directed PRR_{n2n} from sensor i to sensor j. The entry $q_{i(n+1)}$ represents the PRR_{n2n} from sensor i to the sink. In addition, PRR_{n2n} is non-reflexive with q_{ii}=0. The WSN can be represented as a directed graph G=$\{V, E\}$ with sensors V=$\{v_1, \ldots, v_{n+1}\}$, and edges E=$\{(v_i, v_j)|q_{ij} > 0\}$. A binary routing table R=r_{ip} of equal size to Q_{PRR} assigns each sensor i to a single receiver p (*parent*) with r_{ip}=1. A routing table is called *valid* if it fulfills the following constraints:

$$\sum_{p=1}^{n+1} r_{ip} = 1 \quad (1) \qquad r_{ii} = 0 \quad (2) \qquad r_{ip} = 1 \Rightarrow (v_i, v_p) \in E \quad (3)$$

for $i \leq n$ and $p \leq n$+1. Hence, any sensor with the exception of the sink, has exactly one parent (1), no sensor is its own parent (2), and a directed route requires a link (3). The routing table results in a routing tree with each path terminating at the sink. A TDMA schedule frame F=f_{si} is a binary table of size $t \times n$ which represents a recurring scheme of t time-slots of equal length. If f_{si}=1, then sensor i is allowed to transmit a packet every ($s \mod t$)th time-slot. Considering both primary and secondary transmission conflicts, the following constraints must hold for F to be *collision-free* (detailed description of conflicts in, e.g., [7]):

$$f_{si} = 1 \Rightarrow f_{sj} q_{ji} = 0 \quad (4) \qquad \sum_i f_{si} q_{ij} \leq 1 \quad (5)$$

for $s \leq t$, $i \leq n$, and $j \leq n$+1. Thus, a sensor cannot receive at the same time as it transmits (4), and a sensor may not be in the range of two transmitting sensors (5). In addition, we require the probability of the event A=*'All packets scheduled in one frame arrive at the sink in the same frame'* to be above zero with $P_{R,F}(A)$>0, in order to exclude routing-schedule pairs that imply starvation for a subset of sensors. A solution fulfilling all constraints is called *successful*. Evolved lower bounds either guarantee that QoS constraints are met or that the topology requires adjustment.

IV. Approach

GAs are population-based algorithms that trigger an artificial evolution in order to find solutions to hard optimization problems [11]. Individuals are encoded as *genomes* which translate into *phenomes*; solutions to the problem. From generation to generation, the GA clones and manipulates the *fit* genomes using recombination and mutation operators. Fitness is defined as the ability of phenomes to survive in the environment, assessed by the *fitness function*. A selection mechanism enforces the *survival of the fittest*. In relation to the success of a GA, the appropriate choice of fitness function, genome representation, and operators that both drive diversity and preserve successful patterns (*building blocks*) are vital in order for the population to evolve into fit regions of the search space. ¡Existing work on schedule optimization surveyed in [6] lists contributions using GA, but no work, to our knowledge, combines QoS and cross-layer optimization.

V. Application Scenario

We investigate sparsely connected multi-hop WSN with 17 sensors collecting mission-critical readings. We created 100 random topologies with 16 sensors and a single sink. Providing QoS guarantees even for small size networks is difficult. The small size allows us to find reference cross-layer solutions, inducing minimum delay for comparison purposes using integer programs. Those solutions do not consider reliability. The topology graphs G=(V, E) are created as follows. For each sensor v_i a number of between 1 and 3, uniformly at random drawn, neighbors from $V \setminus \{v_i\}$ are assigned directed links. Then, for each link, connectivity q_{ij} is assigned uniformly at random from $[0.95, 0.99]$. The PRR_{n2n} levels are assumed to be comprised of interference. In a true scenario they could be lower-bounds, obtained offline.

VI. Algorithm

A. Representation & Constraint Handling

GA require an encoding for both genome and phenome, and a deterministic translation from genome to phenome. A candidate is composed of a routing table R and a schedule frame F. The routing table and schedule are represented as a binary matrix for the genome and phenome, thus allowing for the use of standard GA mating operators. Mating operations are usually performed on the genome, while fitness assessment is exclusively on the phenome. The genome does not have to fulfill any constraints. For highly constrained problems, such as the present, validity for the phenomes must be guaranteed. The two most common approaches to ensure constraint conformity are penalization and repair algorithms. Penalization is not an option in this case because our fitness assessment simulator only has the ability to assess valid solutions. The GA repairs R and F in parallel and assesses them in combination. Figure 1 provides a conceptual map in relation to where and how this fits into the GA framework. Algorithm 1 realizes deterministic schedule repair. The algorithm identifies any conflicts for each slot (line 2) and resolves them by iteratively removing the transmitting sensors with the highest conflict rate (line 4-6)

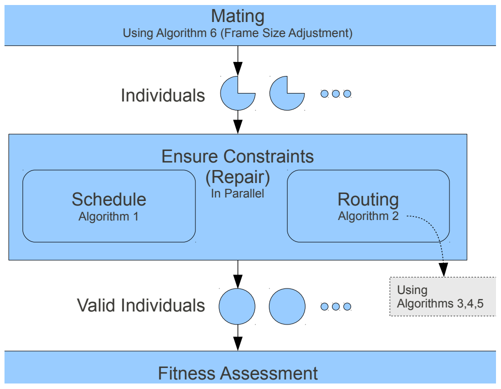

Fig. 1: A conceptual map of the introduced algorithms.

until no conflicts remain. The algorithm is fully parallelizable with regards to the amount of slots t. Algorithm 2 realizes deterministic routing repair, consisting of three steps (lines 1-3): Remove invalid routes in R, recursively connect all sensors to sinks using multi-hop, and remove redundant paths.

```
Algorithm 1: $ScheduleRepair(Q_{PRR}, F)$
Input: Link Matrix $Q_{PRR}$, Genome Schedule $F$
Output: Phenome Schedule $F$
1: for Slot $S$ in $F$ do
2:   $C \leftarrow Conflicts(S, Q_{PRR})$ // t-r, t-t, t-r-t conflicts
3:   while not $C = \emptyset$ do
4:     $i \leftarrow MostConflictingSensor(C)$
5:     $f_{si} \leftarrow 0$
6:     $RemoveConflicts(i, C)$
7:   $AddNonConflictingSenders(S, Q_{PRR})$
```

B. Fitness Function

We consider the constraints (1-5) and $P_{R,F}(A)>0$. Reliability is addressed by maximizing the expected $PRR_{e2e} \in [0, 1[$. PRR_{e2e} estimates are based on a customized number of network simulation repetitions throughout the evolution (here 100). Each simulation involves one candidate with each sensor having an initially specified number of packets waiting for transmission. Latency is addressed by minimizing the schedule size $t \in \{\frac{n}{sinks}, \dots, \frac{n(n+1)}{2}\}$. It is a trivial matter to prove that $\frac{n}{sinks}$ is the lower bound in a single channel WSN. Also, a successful schedule of size $\frac{n(n+1)}{2}$ can always be constructed using exclusive transmission slots starting at the furthest point from the sink. We abandoned obsolete packets so that no packet arriving at the sink is older than t slots. All packets arrive in a time frame with probability PRR_{e2e}. The GA attempts to minimizes fitness function (6), by integrating reliability and latency.

$$f(R, F) = size(F) - PRR_{e2e}(R, F) \tag{6}$$

C. Dynamic Frame Size Adjustment

The initial schedule size and how it changes throughout evolution when fitness is promising (poor) must also be decided. Initial schedule length is defined by the sum of hops to a sink for all sensors given each candidates, repaired initial routing table. R and F are randomly instantiated from $\{0, 1\}$. Algorithm 6 ensures that, during evolution, the offspring of successful schedules is shortened, while for non-successful

```
Algorithm 2: $RoutingRepair(Q_{PRR}, R)$
Input: Link Matrix $Q_{PRR}$, Genome Routing $R$
Output: Phenome Routing $R$
1: $RemoveDeadLinks(Q_{PRR}, R)$
2: $Connect(sink, \emptyset, Q_{PRR}, R)$ // Alg. 3
3: $RemoveRedundant(sink, \emptyset, Q_{PRR}, R)$ // Alg. 4
```

```
Algorithm 3: $Connect(parent, Marked, Q_{PRR}, R)$
Input: Sensor $parent$, Marked Sensors $Marked$, Link Matrix
    $Q_{PRR}$, Routing $R$
1: for $child \in Children(parent, Q_{PRR})$ do
2:   if not $child \in Marked$ then
3:     if not $IsConnected(child, Q_{PRR}, \emptyset, Marked)$ then
4:       $r_{child,parent} \leftarrow 1$
5:     $Marked \leftarrow Marked \cup \{child\}$
6:     $Connect(child, Marked, Q_{PRR}, R)$
```

```
Algorithm 4: $RemoveRedundant(parent, Marked, Q_{PRR}, R)$
Input: Sensor $parent$, Marked Sensors $Marked$, Link Matrix
    $Q_{PRR}$, Routing $R$.
1: for $child \in Children(parent, R)$ do
2:   if not $child \in Marked$ then
3:     for $altParent \in Parents(child, Q_{PRR})$ do
4:       if $altParent \neq parent$ then
5:         $r_{child,altParent} \leftarrow 0$
6:     $Marked \leftarrow Marked \cup \{child\}$
7:     $RemoveRedundant(child, Marked, Q_{PRR}, R)$
```

```
Algorithm 5: $IsConnected(sensor, Q_{PRR}, Path, Marked)$
Input: Sensor $sensor$, Marked Sensors $Marked$, Path $Path$, Link
    Matrix $Q_{PRR}$.
1: if $sensor \in Path$ then
2:   return false
3: if $sensor \in Marked$ or $sensor = n$ then
4:   return true
5: $Path \leftarrow Path \cup \{sensor\}$
6: for $parent \in Parents(sensor, Q_{PRR})$ do
7:   if $IsConnected(parent, Q_{PRR}, Path, Marked)$ then
8:     return true
9: return false
```

ones it is prolonged. For offspring shorter than their parents, slots are randomly removed from the frame. Prolonged offspring receive random repetitions of slots in order to fill up the frame. The proportional change X in algorithm 6 is for each individual drawn uniformly at random.

```
Algorithm 6: $DynamicFrameSizeAdjustment(I)$
Input: Parent Phenome $I$=$(R, F)$
Output: offspring frame size
1: if $P_{R,F}(A)$ then
2:   return $size(F) - X \in \{0, \dots, \lceil \frac{size(F)-l_b}{2} \rceil\}$
3: else
4:   return $size(F) + X \in \{0, \dots, \lceil \frac{u_b-size(F)}{2} \rceil\}$
```

VII. Simulations and Results

All simulations were conducted in a Java GA simulator on a Dell T3500 Intel Xeon Quad Core with 6 GB RAM. A parameter tuning that, due to space restrictions, cannot be explained in detail, has led to the chosen settings in Table I. The GA is evaluated 30 times on each topology using the fitness function (6). Study 1 explores the best-effort

Operators	
VerticalSinglepoint-SchedulingCrossover	(VSSC)
ToggleRouting-Mutation	(TRM)
ToggleScheduling-Mutation	(TSM)

Parameter	Value
PopSize (EliteCount)	26(5)
Termination (Time)	$5s$
Selection(Selected)	Sigma(3)
VSSC.Probability	0.81
TSM.Probability	0.01
TRM.Probability	0.025

TABLE I: The best identified setting for the Lamarckian GA.

capabilities, while in study 2 a latency bound of 30% distance to the optimum (dto) was added in order to emulate the more realistic case of strict submission deadlines. QoS, in terms of reliability and latency is assessed after five seconds. The GA results are compared to the features of the reference solutions. Table II summarizes the numeric results. Gurobi 5.5, a state-of-the-art solver, required an average of 34 minutes to identify the reference solutions. For study 1, the expected reliability has a mean PRR_{e2e} of 0.951 while the latency is 4.5% from the optimum. In each run of the second study, solutions with 30%

	Delay (dto) %	$\mathbf{PRR_{e2e}}$	Time (s)
Reference	0	0.920 ± 0.013	2039 ± 13692
Study 1	4.5 ± 5.8	0.951 ± 0.012	**5**
Study 2	**30**	0.997 ± 0.006	**5**

TABLE II: The numerical results and reference solutions.

dto latency were found. An accumulated 0.32% of the packets are expected not to arrive in time as compared to 8.02% for the solutions with optimal latency.

VIII. Discussion

The GA continuously evolves valid lower bounds with QoS features that can be monitored until convergence is achieved. Within 5 seconds, high reliability, low latency solutions could be identified for all topologies. In balance with the natural assessment, we observe that retaining a stable reliability while decreasing latency is the biggest challenge and it is not reliability or latency in isolation. Still, assuming routing to be given, however, it is still the case that identifying optimal latency in isolation is NP-hard. The application-level QoS depends highly on the MAC layer, making it hard to find optimally matching solutions without the consideration of cross-layer optimization. Consideration being given to the reliability appears to boost the optimization. A possible explanation is that the evolved solutions that do increase reliability are more likely to produce successful offspring with shorter schedules. We speculate that the inclusion of reliability could also boost other heuristics. A centralized approach on small size networks raises questions with regards to scalability and complexity. Because GAs are non-deterministic approximation algorithms, worst case guarantees cannot be made. However, the GA has a population of size p, running for n consecutive generations. This results in a fitness evaluation complexity of n, as all p individuals can be evaluated in parallel. Since all network simulations (not merely individuals) can run simultaneously, the complexity of the fitness function is equal to the complexity of one run of the network simulator NS, leading to a total expected time of $n * time(NS(1))$. Routing and scheduling repair can be executed in parallel. The execution time should not be an issue even for larger instances above 100 sensors, but the effects on the result quality have to be investigated. The stochastic assessment coupled with a constant challenging of the candidate, create robust reliability levels. The algorithm evolves an average of 177 generations per second. The proposed method is not limited to use for converge-cast scenarios, and could be extended for multi-path routing or multi-channel transmission which would cause it to become more closely compatible with existing WSN protocol stacks, such as WirelessHART.

IX. Conclusions

We introduced a MAC and routing cross-layer approach to QoS assessment and optimization in a WSN. The approach can be used both as a tool for rapid assessment of expected QoS in deployed networks, and the finding of cross-layer routing and TDMA schedules for utilization in the network. We have investigated two modi; best-effort and latency constraint. One of the major advantages of GA is that a change in fitness function is relatively simple. Thus, extending them for other QoS features will also prove to be rather simple. The simulation results suggest that substantial gain in reliability can be achieved, while only a small quality loss in latency is introduced, in comparison to that for hard-to compute minimal length TDMA schedule routing-tree pairs.

References

[1] J. Akerberg, M. Gidlund, and M. Bjorkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," in *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on*, 2011, pp. 410–415.

[2] 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. 14, no. 2, pp. 240–264, 2012.

[3] L. D. Mendes and J. J. Rodrigues, "A survey on cross-layer solutions for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 523 – 534, 2011.

[4] T. Lennvall, S. Svensson, and F. Hekland, "A comparison of wirelesshart and zigbee for industrial applications," in *Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on*. IEEE, 2008, pp. 85–88.

[5] "Wirelesshart technology," HART communication foundation, Tech. Rep., Dec 2009.

[6] S. Ergen and P. Varaiya, "Tdma scheduling algorithms for wireless sensor networks," *Wireless Networks*, vol. 16, pp. 985–997, 2010.

[7] P. Djukic and S. Valaee, "Delay aware link scheduling for multi-hop tdma wireless networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 870–883, Jun. 2009.

[8] P. Suriyachai, J. Brown, and U. Roedig, "Time-critical data delivery in wireless sensor networks," in *Proceedings of the 6th IEEE international conference on Distributed Computing in Sensor Systems*, 2010, pp. 216–229.

[9] S. Munir, S. Lin, E. Hoque, S. M. S. Nirjon, J. A. Stankovic, and K. Whitehouse, "Addressing burstiness for reliable communication and latency bound generation in wireless sensor networks," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN '10, 2010, pp. 303–314.

[10] 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, 2006.

[11] K. Man, K. Tang, and S. Kwong, "Genetic algorithms: concepts and applications [in engineering design]," *Industrial Electronics, IEEE Transactions on*, vol. 43, no. 5, pp. 519 –534, oct 1996.