

Novel Load Balancing Scheduling Algorithms for Wireless Sensor Networks

Endre László, Kálmán Tornai, Gergely Treplán
 Faculty of Information Technology
 Pazmany Peter Catholic University
 Budapest, Hungary
 lasen@digitus.itk.ppke.hu,
 tornai.kalman@itk.ppke.hu, trege@itk.ppke.hu

János Levendovszky
 Department of Telecommunications
 Budapest University of Technology and Economics
 Budapest, Hungary
 levendov@hit.bme.hu

Abstract—In this paper, optimal scheduling mechanisms are developed for packet forwarding in wireless sensor networks, where clusterheads are gathering information. The objective is to monitor real-life processes for a given time interval and forward packets with minimum loss probabilities to the base station. In order to achieve this objective we develop an optimal scheduling algorithm, which determines the time slots in which packets must be sent by the nodes. The scheduling algorithm, on the one hand, guarantees that all the packets will be sent within a predefined time window and thus meeting delay constraints and, on the other hand, it provides uniform packet loss probabilities for all the nodes. The algorithm we propose is capable of providing optimal scheduling with given constraints and guarantees balanced load in polynomial time.

Index Terms—load balancing, scheduling, wireless sensor networks (WSN), numerical optimization

I. INTRODUCTION

Data gathering from a set of sensor nodes to a Base Station (BS) by using a cluster-based routing topology is commonly used in wireless sensor networks (WSNs) [1], [2]. In this kind of networks tiny sensor nodes communicate in short distances and collaboratively work to fulfill the application specific objectives of WSN. Many of the envisioned applications involve the collection of bursty data traffic generated by events, which are to be delivered to the BS as quickly and as reliably as possible in order to recognize emergency situation. In these applications packet delay and packet loss probability are of crucial importance [3], [4].

Resources (energy, time, bandwidth) are limited in WSN applications, hence cross-layer optimization are the key approach to minimize the utilization of resources if the Quality of Service (QoS) is prescribed.

Because of these strong limitations, scheduling becomes even more important in WSN to save on energy consumption. In this paper we develop a new scheduling algorithm for packet transfer in WSN, which guarantees reliable information transmission to the BS in terms minimizing the packet loss probabilities. Packet loss occurs when the amount of packets forwarded to the CHs exceed their capacity. Scheduling has been intensively researched in the telecommunication literature [5], [6], [7], however the main focus was on buffered architectures. The problem in our case will be transformed

into a binary matrix optimization. So far, this has been solved by quadratic programming (see [8]). Now a novel solution is presented, which will provide the optimal scheduling matrix in polynomial time. Since clusterhead (CH) based routing is a commonly used solution in WSN (e.g., LEACH protocol [9] or other hierarchical solutions proposed in [10], [11]), we assume that each node can only send one packet to a selected CH at each time instant.

After the system and the application are briefly introduced, the network model and the mathematical model are detailed. The formulation and transformation of the original problem into a matrix optimization task is followed by the performance analysis and the conclusion of our work.

II. SYSTEM DESCRIPTION AND APPLICATION

The algorithm introduced in this paper is concerned with applications when the traffic generated by the sensor nodes are classified into traffic classes (e.g., motion detection, acoustic signals, video signals, ... etc.) The WSN architecture is single-tier clustered with heterogeneous sensors and performing centralized processing [12]. The CHs forward the received packets to the BS over a link with capacity V . As a result V presents a limit to the packets can be collected from nodes in a given time instant. In order to collect all the packets the CH assigns timeslots to the nodes for acquiring their packets. This timeslot assignment is referred as a scheduling. In this scheme, each traffic class has two parameters:

- the amount of packets to be transmitted to the CH;
- the time duration in which this given amount of packets must arrive at the CH.

The target platform is the Berkeley Mica2 mote [13], which is one of the most widely used WSN platforms. The platform has an 8 MHz processor, 4 kB of RAM, 128 kB of flash memory and a 433 MHz wireless radio transceiver. The transfer rate is 38.4 kbps and it is powered by two AA batteries.

TinyOS operating system [14], [15] is designed to be used with networked sensors and it supports the Mica2 platform. TinyOS handles task scheduling, radio communication, clock and timers, ADC and power management.

In the case of Mica2, a fully programmable Media Access Control (MAC) layer is available. A time division multiple access (TDMA) channelization is supported in the case of such simple and energy constrained nodes. Plenty of channel access method is proposed for WSN and there are two common solutions. One solution is the contention based scheme where nodes try to assign the channel randomly, independently from each other, such as ALOHA [16]. ALOHA is improved in many ways (such as B-MAC [17], Z-MAC [18], X-MAC [19]) to guarantee performance metrics required by different applications. Contention based protocols are very important channel access methods in the case of rare but bursty traffic, which is the typical case of event driven applications. On the other hand TDMA-based scheduled solutions could be very efficient if the users are known and fix, the data arrives regularly (e.g.: every one minutes the temperature value must be delivered to the BS) and topology is static. TDMA-based protocol can also save more energy, since each node can stay in sleep mode except for its own slot time. However TDMA-based protocols need synchronization protocols [20] affecting the guard time [21], time slot assignment [22] determining the end to end delay and scheduling. Scheduling protocols choose the order of the data transfer at the different links [23]. In this paper we propose Load Balancing Scheduling (LBS) algorithm, which minimizes the probability of cell loss.

In the following section the mathematical model will be introduced.

III. THE MODEL

The overall number of packets node j wants to send to the CH is denoted by X_j . All of the packets must be sent within a time interval $K_j : K_j > X_j$. The scheduling of packet transmissions by node j can be expressed by a binary vector $c(j)$ of length K_j with weight X_j , where component $c_j(l) \in \{0, 1\}$ indicates whether packet is transmitted or not to the CH at time instance l .

Composing a transmission matrix \mathbf{C} of these vectors, one may seek the optimal matrix (\mathbf{C}_{opt}). To find the optimal matrix, we present a novel approach based on a constructive iteration, solving the problem in polynomial time.

The data acquisition network can be divided into two parts:

- The sensors performing measurements in the different rooms.
- Relay nodes, which are forwarding the packets containing the measurements to the BS.

In Figure 1 the different application classes correspond to different rooms in a building. Each room has different X_j , i.e., one requests a given amount of data from every room. The data is collected by the sensor node from its sensors and stored for transmission. In the beginning of the communication X_j -s are transferred to the sensors nodes, where the calculation of K_j -s are performed based on the battery status and returned to the CH. The optimal scheduling matrix is then constructed by the CH. In each time-slot, V amount of data is sent to the CH corresponding to the l -th column of the scheduling matrix.

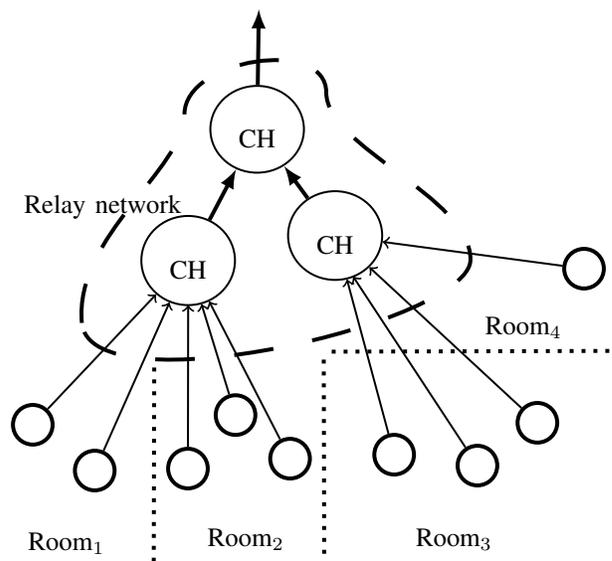


Fig. 1. WSN communication scenario with sensor nodes and CHs

This V amount of data is then sent to the relay nodes of the network in order to forward them to the BS.

The nodes performing the measurements are small size units with a variety of sensors included with strong energy and processing limitations. Energy is used either in stand-by mode or in active mode when they are engaged with radio transmission (packet forwarding). The available energy at node j is denoted by G_j , while the energy consumption in active node is g_{tx} and in stand-by is g_{sleep} . If the node sends X_j amount of packets and it switches off after K_j timeslot, then the overall energy consumption can be given as

$$G_j = (K_j - X_j) \cdot g_{sleep} + X_j \cdot g_{tx}. \quad (1)$$

As G_j is limited (due to the battery capacity) based on this expression one can determine what is the maximal K_j needed to transmit a given amount packets by the sensor node. As far as the relay nodes are concerned, they have more energy than the sensor nodes. Each room has a relay node, which acts as a CH. The CH can only receive V number of packets in a time frame due to its capacity. In the case of TDMA, V is the number of slots accommodated in a time frame. The task of scheduling is to assign time slots to the sensor nodes in such a way that the X_j number of packets are sent in K_j time and the probability of packet loss (in a given time slot more than V sensor nodes are sending packets to the CH) must be uniformly distributed.

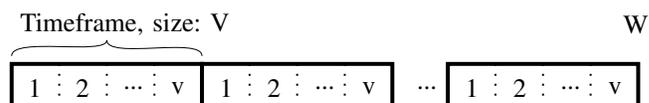


Fig. 2. Typical structure of TDMA timeframes

The number of timeframes available in a relay node is denoted by W . The number of available timeslots in each

timeframe is denoted by V . These slots are summarized on Figure 2. In a TDMA protocol each node has a dedicated timeslot in every timeframe wherein it may send its packet to its parent node.

A. Mathematical model

Let us assume that there are J number of nodes transmitting packets to a specific cluster head. The capacity of the cluster head in a time frame is denoted by V . This capacity means the amount of packet can be received in one time frame, the number of time slots. The amount of packets to be sent by node j is denoted by X_j , while the time constraint in which the transmission is to be finished is denoted by K_j . X_j is initially defined and using the equation (1) the K_j can easily be calculated.

The time is measured in discrete units thus $K_j, j = 1, \dots, J$ are assumed to be integers. The scheduling of node j is represented by a binary vector $\mathbf{c}_j \in \{0, 1\}^{K_j}$ where if $\mathbf{c}_j(l) = 1$ then a packet is sent to the cluster head at time instant l . The scheduling matrix \mathbf{C} can be constructed from vectors $\mathbf{c}_j, j = 1, \dots, J$, which form the row vectors of \mathbf{C} and the number of columns is taken as $L = \arg \max_j K_j$. For example in the case of $J = 6, X_1 = 5, X_2 = 5, X_3 = 4, X_4 = 3, X_5 = 3, X_6 = 7, K_1 = 6, K_2 = 8, K_3 = 7, K_4 = 6, K_5 = 4$, and $K_6 = 8$ one specific scheduling matrix looks as follows:

$$\mathbf{C}_{\text{valid}} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

In this example the capacity of the CH was $V = 4$, and as it can be seen the number of ones are balanced. The following example for \mathbf{C} matrix is not valid and unbalanced, and this scheduling overloads the CH.

$$\mathbf{C}_{\text{invalid}} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

The aggregated number of incoming packets in the cluster head at time instant l is given as $\sum_{j=1}^J \mathbf{c}_j(l)$ being the weight of the l^{th} column vector in matrix \mathbf{C} . The cost of reception (which needs the power of CHs) is proportional to the number of received packets at time instant l expressed as $a_l := \sum_{j=1}^J \mathbf{c}_j(l)$. We seek an optimal matrix \mathbf{C}_{opt} where the aggregated number of incoming packets are balanced with respect to time. More precisely, we try to achieve that nodes schedule their packet transmission in such a way that each time instant the cluster heads receive more or less the same number

of packets. This objective can be formulated as follows:

$$\mathbf{C}_{\text{opt}} : \min_{\mathbf{C}} \sum_{l=1}^L \sum_{k=1}^L \left(\sum_{j=1}^J C_{j,l} - \sum_{j=1}^J C_{j,k} \right)^2. \quad (2)$$

The constraint can be expressed as $\sum_{l=1}^L C_{j,l} = X_j, j = 1, \dots, J$ and if the last nonzero component of row j is at location M_j , such that $M_j : \sum_{l=1}^L C_{j,l} = X_j, j = 1, \dots, J$ then $M_j \leq K_j, j = 1, \dots, J$. We seek the balanced solution by optimizing (2).

Furthermore we deal with the assumption that there is a loss in the transfer of packets to the CH because of the overload of the link. Namely, there exists no scheduling matrix for, which the aggregated load does not exceed the throughput V of the CH node, i.e., for every optimal scheduling matrix \mathbf{C}_{opt} there exists at least one time instance when the aggregated load exceeds the throughput:

$$\forall \mathbf{C}_{\text{opt}} : \exists k, V < \sum_{j=1}^J X_j(k) \quad (3)$$

This assumption is important because it means that the available bandwidth is fully utilized.

B. The scheduling protocol

In order to optimize matrix \mathbf{C} we need parameters $X_j, K_j, j = 1, \dots, J$. In the first stage of the protocol we define the amount of packets we want to acquire data from the different sensor nodes (e.g. we want to get information about the status of the adequate room), which will specify parameters $X_j, j = 1, \dots, J$. In the forthcoming discussion we define a communication protocol. Therefore, initially we assume that in each room $X_j, j = 1, \dots, J$ are given.

- 1) The nodes are synchronized using the Reference Broadcast Synchronization (RBS [24]) protocol, in order to use the TDMA based package scheduling.
- 2) The values of X_j are sent to the sensor nodes in order to specify the amount of packets to obtain a certain amount of information about the status of room containing node j .
- 3) Each sensor node determines the amount of timeframes ($K_j, j = 1, \dots, J$, one timeslot in each timeframe), wherein the X_j packets can be transmitted. In other word, by using equation (1) they calculate $K_j, j = 1, \dots, J$. This information is sent back to the CH.
- 4) Having $X_j, K_j, j = 1, \dots, J$ the CH will determine the optimal scheduling matrix \mathbf{C}_{opt} .
- 5) If at any given timeslot an overload occurs (the weight of a given column vector exceeds V) then some randomly selected packets are discarded subject to uniform distribution. In this way, the fair treatment is provided for all sensor nodes.
- 6) Each node will receive the corresponding row vector of the optimized scheduling matrix \mathbf{C}_{opt} .
- 7) In operation the nodes are transmitting packets to the CH only in those timeslot, which are indicated in the received row vector.

In the next section, the optimization of matrix \mathbf{C} is discussed by using a constructive polynomial algorithm named LBS.

IV. SOLUTION BY POLYNOMIAL ALGORITHM: LBS ALGORITHM

In this section a new algorithm is presented, which aims at balancing the number of “1”-s in every column vector of matrix \mathbf{C} . The forthcoming algorithm is motivated by a simple consideration that if the scheduling of the load can be distributed uniformly over a given time interval, then the packet loss is reduced to the minima. Any deviation from this uniformity may cause loss to a given time instant, while in other time slots under-utilization occurs.

Let the the matrix \mathbf{C} initially be filled in with zeros. We modify the matrix as $C_{l,k} = 1$ if $k < K_l$ and $l = \arg \min_j K_j$. In this way, the algorithm will place L number of “1”-s in the matrix at the first L steps. One may see, that in each step that specific node is selected, which has the smallest remaining lifespan. After L steps the algorithm steps to the end of the matrix and the previous steps this construction procedure is repeated. Note that this algorithm will never put “1” in a place where the constraints forbid it ($X_j < K_j$). The algorithm terminates when all the prescribed $X_j, j = 1, \dots, J$ number of packets are scheduled.

This method named LBS algorithm (Algorithm 1) can formally be described as follows:

Algorithm 1 LBS algorithm

Require: $\forall i = 1 \dots J : K_i, X_i$
 $L \leftarrow \max_j K_j$
 $\mathbf{C} \leftarrow \mathbf{0}_{J \times L}$
 $S \leftarrow \sum_j X_j$
while $S > 0$ **do** {Number of unscheduled packets}
 for $l \leftarrow L$ **to** 1 **do**
 Find $\arg \min_i K_i$ where $(l < K_i) \wedge (X_i > 0)$
 $S \leftarrow S - 1$
 $X_i \leftarrow X_i - 1$
 $\mathbf{C}_{i,l} \leftarrow 1$
 end for
end while

Let us analyze the complexity of the newly proposed LBS algorithm. We put the number of $\sum_{j=1}^J X_j$ “ones” into the matrix and this can be upper bounded by JL . There is a minimization phase in each insertion, which be evaluated in time J . Thus, the overall complexity is as follows

$$J \cdot \sum_{j=1}^J X_j \sim \mathcal{O}(J \cdot (J \cdot L)) \sim \mathcal{O}(J^2 \cdot L). \quad (4)$$

This complexity is between the cubic and quadratic time taking account the following assumption: $J < L$ therefore this algorithm runs in polynomial time.

The newly presented LBS algorithm gives optimal solution to (2) and runs in polynomial time. In the next section extensive simulations demonstrate the performance of the method.

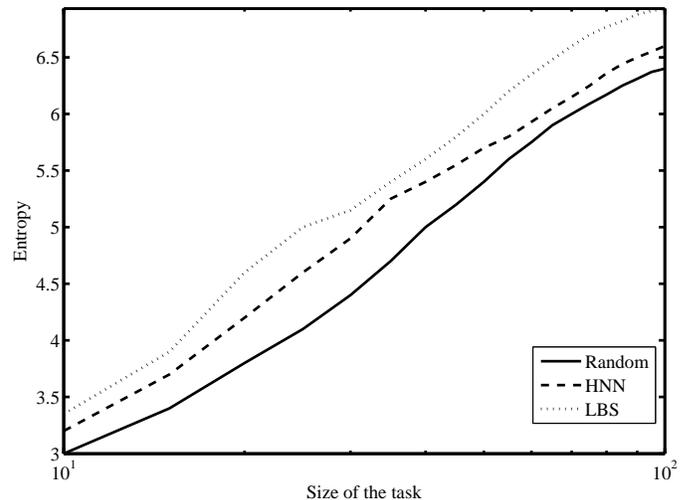


Fig. 3. Entropy of solutions achieved by different methods

V. PERFORMANCE ANALYSIS

In this section, we compare the optimal solution obtained by exhaustive search to the one achieved by the LBS algorithm and to the one achieved by a random strategy. The simulations were carried out for $J = 20$ nodes and the corresponding $X_j, j = 1, \dots, J$ and $K_j, j = 1, \dots, J$ constraints have been chosen randomly in the range of $X_j \in [5, 100]$ and $K_j \in [5, 100]$. The results have been evaluated after selecting several constraints randomly, then running the simulations and taking the average error of the solutions achieved by the methods and by exhaustive search. The random method and the exhaustive search are the reference solutions in order to provide a better comparison of this new LBS method.

We took into consideration the properties of the platform specified in Section II in the simulations we performed.

The obtained results are depicted by the Figure 3. Analyzing the result on the figure, one may note that the solution provided by the LBS algorithm is possible as good as the results provided by the Exhaustive search. The Random strategy performs more worse than the LBS algorithm or the Exhaustive search. The figure shows the entropy of the solutions. One may note the horizontal axis scaled logarithmically. (The exhaustive search is not visible as the results of the LBS algorithm fully overlaps it.)

The previous and following graphs contain the results of a Hopfield Neural Network (HNN), our previous solution to solve the specified problem. Details about the HNN may be found in [8] and [25]. The results of the HNN only slightly differs from the optimum.

The running time of the enumerated methods are compare to each other in Figure 4. The LBS algorithm, the HNN and the random strategy provide its solutions in polynomial time opposed to to the exponential complexity of the exhaustive search method. (The exhaustive search due to the exponential complexity does not appear on the previous graph.) The LBS

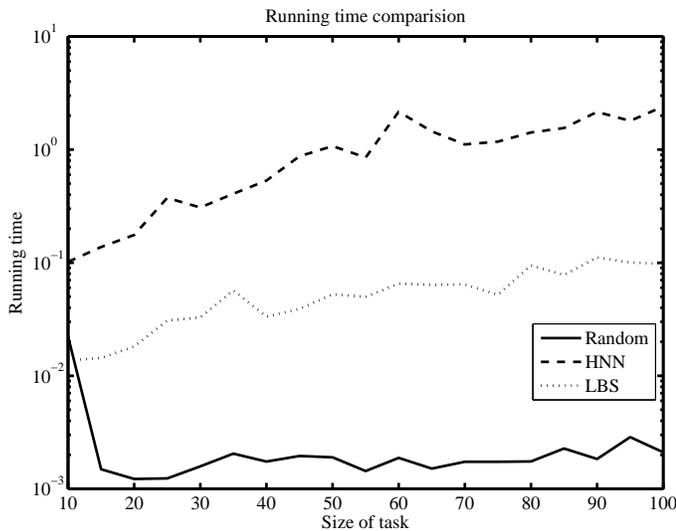


Fig. 4. Running time of solutions achieved by different methods

algorithm is faster than the HNN recursion, the reason are detailed:

- The random strategy generates a random matrix keeping the constraints $X_j, K_j : \forall 1 \dots J$. The complexity of this generation is

$$\mathcal{O}(J \cdot L)$$

- The theoretical complexity of the LBS algorithm is (as detailed previously):

$$\mathcal{O}(J^2 \cdot L)$$

- The time and steps of convergence for the discrete HNN using the strong Lyapunov criteria could be proven as follows:

$$\mathcal{O}(J^2 \cdot L^2)$$

- The exhaustive search uses huge search space and elects the optimal scheduling matrix from all possible C matrix. Therefore its complexity:

$$\mathcal{O}(2^{J \cdot L})$$

Therefore we expect the results depicted by the Figure 4. (The running time is measured in seconds and the vertical axis is scaled logarithmically.) We repeated our tests several times but not enough times to provide monotone variation as one may expect. Certainly these results can be extrapolated.

In order to measure how balanced a solution is, we introduce the entropy of the weight distribution of the columns in matrix C as follows

$$H(\mathbf{p}) = \sum_{k=1}^L -p_k \ln p_k, \tag{5}$$

where

$$p_k = \frac{\sum_{j=1}^J C_{jk}}{\sum_{k=1}^L \sum_{j=1}^J C_{jk}}$$

This entropy value provides a heuristic metric to describe the measure of equalization of the packet loads in the different solutions.

Table I demonstrates that the solution provided by LBS algorithm has the highest weight entropy (i.e., the most uniform weight distribution of the columns), hence it fulfills the constraint related to weight balancing.

| Method | Entropy |
|-------------------|---------|
| Random Strategy | 5.45 |
| HNN | 5.62 |
| LBS algorithm | 5.89 |
| Exhaustive Search | 5.89 |

TABLE I
ENTROPY OF WEIGHT DISTRIBUTION

Figure 5 shows the advantages of the LBS algorithm, since the number of packets hitting the CH at in time frames are equalized.

One may also see that if a capacity V is given (in this example let $V = 9$) then the scheduling methods provided either by the Random Strategy or by the HNN will suffer from packet loss, as the CH cannot receive all the packets in the given time instances. This effect is very sharp if the timeframes are less than 25. In the case of LBS algorithm, or Exhaustive Search there is no packet loss even in these cases due to more uniform distribution of the load.

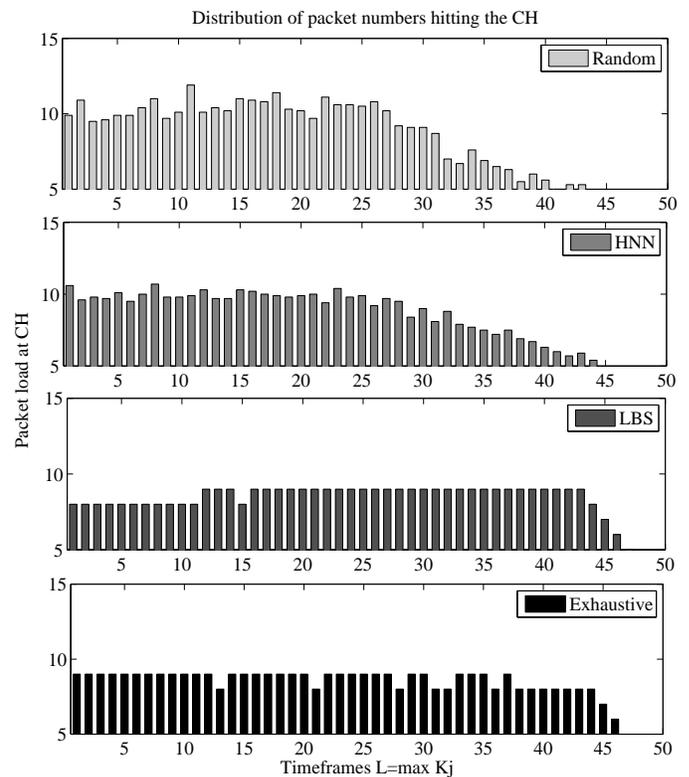


Fig. 5. Received packets in timeframes

The random strategy follows from a simple algorithm of constructing a scheduling matrix. In each row of the matrix the X_j number of “1”-s are randomly placed within the K_j time limit. This method is used for comparison, it serves as a “dummy” model.

VI. CONCLUSION

In this paper, new methods were proposed to provide optimal scheduling for packet-transmissions in WSN. The objective of optimal scheduling was defined by the Balanced Cost objective function. It is noteworthy, that the newly introduced methods yields always a valid scheduling matrix, as opposed to the HNN or random scheduling, which may yield invalid matrices. The algorithm introduced above is capable of providing optimal scheduling with given constraints and guarantees balanced load. It runs in polynomial time and yield valid solution therefore it is possible to use in real-life WSN data acquisition applications.

REFERENCES

- [1] A. Rogers, D. D. Corkill, and N. R. Jennings, “Agent technologies for sensor networks,” *IEEE Intelligent Systems*, vol. 24, pp. 13–17, 2009.
- [2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [3] C.-Y. Chong and S. P. Kumar, “Sensor networks: evolution, opportunities, and challenges,” *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.
- [4] P. Inverardi and L. Mostarda, “A distributed monitoring system for enhancing security and dependability at architectural level,” pp. 210–236, 2007.
- [5] O. Aumagie, E. Brunet, N. Furmento, and R. Namyst, “NewMadeleine: a Fast Communication Scheduling Engine for High Performance Networks,” RR-1421, Tech. Rep., 2007.
- [6] H. Choi, J. Wang, and E. A. Hughes, “Scheduling for information gathering on sensor network,” *Wirel. Netw.*, vol. 15, no. 1, pp. 127–140, 2009.
- [7] H. Li, P. Shenoy, and K. Ramamritham, “Scheduling messages with deadlines in multi-hop real-time sensor networks,” in *RTAS '05: Proceedings of the 11th IEEE Real Time on Embedded Technology and Applications Symposium*. IEEE Computer Society, 2005, pp. 415–425.
- [8] J. Leventovszky, E. Laszlo, K. Tornai, and G. Treplan, “Optimal pricing based resource management,” in *International Conference on Operations Research Munich 2010*, September 2010, p. 169.
- [9] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*. IEEE Computer Society, 2000.
- [10] S.-H. Cha and M. Jo, “An energy-efficient clustering algorithm for large-scale wireless sensor networks,” in *GPC'07: Proceedings of the 2nd international conference on Advances in grid and pervasive computing*. Springer-Verlag, 2007, pp. 436–446.
- [11] K. Wu, C. Liu, Y. Xiao, and J. Liu, “Delay-constrained optimal data aggregation in hierarchical wireless sensor networks,” *Mob. Netw. Appl.*, vol. 14, no. 5, pp. 571–589, 2009.
- [12] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, “A survey on wireless multimedia sensor networks,” *Comput. Netw.*, vol. 51, pp. 921–960, March 2007.
- [13] CAPSIL. (2010) Mica2 technical parameters. [Online]. Available: <http://www.capsil.org/capsilwiki/index.php/MICA2>
- [14] TOSWiki. (2010) Tinyos introduction. [Online]. Available: <http://docs.tinyos.net/index.php/FAQ>
- [15] P. Levis. (2006) Tinyos programming. [Online]. Available: <http://www.tinyos.net/tinyos-2.x/doc/pdf/tinyos-programming.pdf>
- [16] N. Abramson, “The aloha system - another alternative for computer communications,” in *Proceedings of Fall Joint Computer Conference, AFIPS Conference*, 1970.
- [17] J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, ser. SenSys '04. ACM, 2004, pp. 95–107.
- [18] I. Rhee, A. Warrier, M. Aia, and J. Min, “Z-mac: a hybrid mac for wireless sensor networks,” in *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, 2005, pp. 90–101.
- [19] M. Buettner, G. V. Yee, E. Anderson, and R. Han, “X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks,” in *Proceedings of the 4th international conference on Embedded networked sensor systems*, ser. SenSys '06. ACM, 2006, pp. 307–320.
- [20] J. Elson and K. Römer, “Wireless sensor networks: a new regime for time synchronization,” *SIGCOMM Comput. Commun. Rev.*, vol. 33, pp. 149–154, January 2003.
- [21] M. Schuts, F. Zhu, F. Heidarian, and F. W. Vaandrager, “Modelling clock synchronization in the chess gmac wsn protocol,” in *Proceedings Workshop on Quantitative Formal Methods: Theory and Applications*, ser. QFM'09, vol. 13, 2010.
- [22] I. Rhee, A. Warrier, J. Min, and L. Xu, “Drand: Distributed randomized tdma scheduling for wireless ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 8, pp. 1384–1396, 2009.
- [23] S. Cui, R. Madan, A. J. Goldsmith, and S. Lall, “Joint routing, mac, and link layer optimization in sensor networks with energy constraints,” in *In Proceedings IEEE ICC 2005*, 2005, pp. 725–729.
- [24] S. Ganeriwal, R. Kumar, and M. B. Srivastava, “Timing-sync protocol for sensor networks,” in *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM New York, 2003, pp. 138–149.
- [25] J. Leventovszky, K. Tornai, G. Treplán, and A. Oláh, “Novel load balancing algorithms ensuring uniform packet loss probabilities for WSN,” in *2011, IEEE 73rd Vehicular Technology Conference, IEEE VTC-Spring*, Accepted in January 2011.