

Optimization of Task Allocation in a Cluster-Based Sensor Network

Mohamed Younis^{*}, Kemal Akkaya and Anugeetha Kunjithapatham

Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
1000 Hilltop Circle
Baltimore, MD 21250
{younis, anu1, kemal1}@cs.umbc.edu

Abstract

Sensor networks have recently gained a lot of attention from the research community. Sensors are significantly resource-constrained devices and last till the depletion of their batteries. Sensor networks typically have a large number of nodes. To ensure scalability sensor networks are often partitioned into clusters, each managed by a cluster head (gateway). Efficient management of a sensor network for extending the lifetime of the network is among the prominent areas of research in this domain. While most of the previous research focused on the optimal use of sensor's energy, very little attention has been paid to the efficiency of energy usage at the gateway. Tasks need to be allocated to gateways in such a way that maximizes the life of these cluster-heads and eventually the whole network. In this paper, we present an optimization scheme for task allocation to gateways. The task allocation problem is modeled as a zero-one nonlinear program. Simulation results show that substantial energy savings can be obtained with the proposed method.

1. Introduction

Over the last few years, the design of sensor networks has gained increasing importance due to their potential for some civil and military applications. Each sensor is capable of detecting ambient conditions such as temperature, sound, or the presence of certain objects. A network of sensors can gather meteorological variables such as temperature and pressure in order to forecast harsh natural phenomena. In disaster management situations such as fires, sensor networks can be used to selectively map the affected regions directing the nearest emergency response unit to the fire. In military situations, sensor networks can perform surveillance missions by detecting moving targets, chemical gases, or presence of micro-agents.

Sensors are generally equipped with data processing and communication capabilities. The sensing circuit measures parameters from the environment surrounding the sensor and transforms them into an electric signal. Processing such a signal reveals some

^{*} Mohamed Younis is the corresponding author. He can be reached by phone at (410) 455-3968, by FAX at (410) 455-3969 and by e-mail at younis@cs.umbc.edu.

properties about objects located and/or events happening in the vicinity of the sensor. The sensor periodically sends such sensed data, usually via radio transmitter, to a command center either directly or through a data concentration center (a gateway). The gateway can perform fusion of the sensed data in order to filter out erroneous data and anomalies and to draw conclusions from the reported data over a period of time. For example, in a reconnaissance-oriented sensor network, sensor data indicates detection of a target while fusion of multiple sensor reports can be used for tracking and identifying the detected target [1].

Deployment of large number of unattended sensor nodes and the underlying network architecture, which will efficiently enable the cooperation of those nodes, has become one of the challenging areas in wireless sensor network research. Since energy is a crucial constraint on sensor nodes, the main aim in sensor network management is to conservatively consume the sensor's energy in order to increase the lifetime of the whole network [2][3][4][5]. One of the possible approaches to the problem of deploying the sensor nodes is to employ network clustering in order to distribute the load evenly and efficiently throughout the whole network [4]. Such cluster-based architecture assigns for each cluster a gateway, which are less energy constrained than sensors. Each gateway is responsible for a particular cluster and interacts with command node and collaborates with other gateways on executing required missions.

Given their geographical proximity to the sensor nodes, the gateways typically perform most of the data fusion and network management activities. Although gateways are relatively less energy constrained compared to sensors, excessive computation and communication load can quickly consume the gateways' batteries and make them a cause for shortening the network lifetime. Therefore, efficient distribution of load among gateways is crucial in order to extend the life of the gateway's battery. On the other hand, most of the data collection tasks are time sensitive. For example accurate target tracking requires sampling at a rate proportional to the target speed. Therefore, the task allocation algorithm must take into consideration the real-time constraints. Since the end-to-end system response time of distributed applications is affected significantly by inter-task communication, one must account for the effect of delays and precedence constraints imposed by inter-task communication when task-allocation decisions are made.

Although task allocation problem has been studied within the scope of parallel and distributed data processing systems, we are not aware of any work that addresses the issues and constraints found in sensor networks. The contribution of this paper is on the model and algorithm for task allocation and scheduling for cooperative unattended ground sensors network. It is shown in this paper that task allocation to gateways can be modeled as a zero-one nonlinear goal-programming problem. The goal is set for maximize the lifetime of gateways in the network subject to resource and timing constraints imposed by the application environment. The model is studied for a simulated workload. The experimental results demonstrated the effectiveness our approach.

In the balance of this section we describe our cluster-based sensor network architecture and summarize the related work. In Section 2, the task allocation problem is analyzed and mathematical modeling of the problem is described along with the optimization algorithm used. Section 3 discusses the validation efforts and analyzes the simulation results. Finally, section 4 concludes the paper and outline future research directions.

1.1 Sensor Network Architecture

We consider the sensor network architecture depicted in Fig. 1. In the architecture sensor nodes are grouped into clusters controlled by a single command node. Sensors are only capable of radio-based short-haul communication and are responsible for probing the environment to detect a target/event. Every cluster has a gateway node that manages sensors in the cluster. Clusters can be formed based on many criteria such as communication range, number and type of sensors and geographical location [6][7]. In this paper, we assume that sensor and gateway nodes are stationary and the gateway node is located within the communication range of all the sensors of its cluster. Clustering the sensor network is performed by the command node and is beyond the scope of this paper. The command node will inform each gateway node of the ID and location of sensors allocated to the cluster.

Sensors receive commands from and send readings to its gateway node, which processes these readings. Gateways can track events or targets using readings from sensors in any clusters as deemed by the command node. However, sensors that belong to

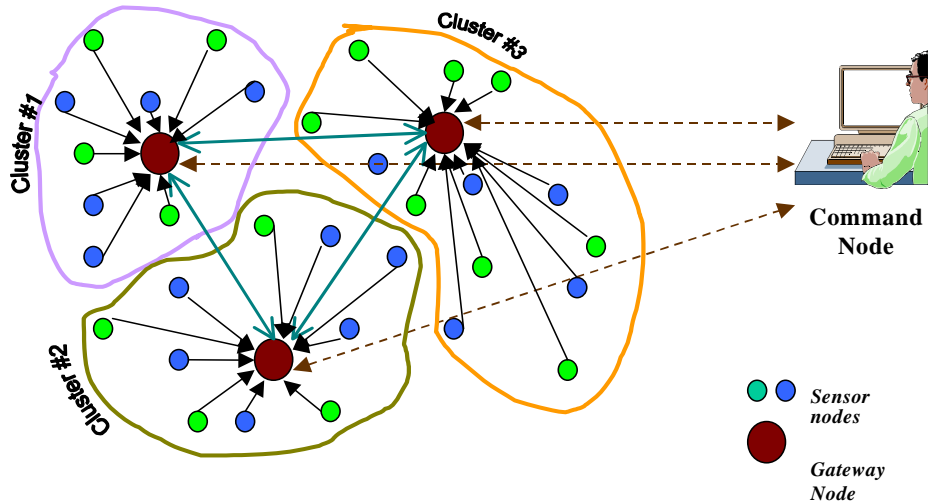


Fig. 1: Multi-gateway clustered network sensors

a particular cluster are only accessible via the gateway of that cluster. Therefore, a gateway should be able to route sensor data to other gateways. Gateway nodes interface the command node with the sensor network via long-haul communication links. The gateway node sends to the command node reports generated through fusion of sensor readings, e.g. tracks of detected targets. The command node presents these reports to the user and performs system-level fusion of the collected reports for an overall situation awareness.

A gateway node can perform target tracking using a set of sensors that belong to another cluster. The increased flexibility allows for efficient and optimized use of the gateway nodes. For example, heavy use of a gateway node can consume most of the node power and thus significantly shorten the life of the gateway. Optimized task allocation among gateways, can expand the life of a certain gateway node by reducing the load (number of tasks) on that node. Although the architecture raises many issues such as cluster formation, cluster-based sensor organization and management, and inter-gateway communication protocol, we only focus on the issue of task arbitration among gateways. Interested readers are referred to [4] for discussion about the network operation within the cluster.

1.2 Related Work

Resource Management typically consists of resource allocation strategy combined with resource usage monitoring. Resource allocation is a classical optimization problem faced in almost all engineering disciplines. Resource consumption models and the mathematical formulation of the allocation strategy are the main variations among resource allocation problems. A bulk of the research related to resource allocation considered centralized resource and task admission control [8][9][10]. Dynamic task assignment to heterogeneous computing resources is studied in [11][12][13]. However, these works lack energy consideration when considering the resources.

Distributed resource management is also modeled as a distributed constrained satisfaction problem [14][15]. Approaches for distributed constrained satisfaction generally requires excessive exchange of messages and fits applications with very dynamic changes in sensor's readings affect the decision of which sensors to turn on next and the required message for applying distributed constraint satisfaction algorithms are needed by the application anyway. However requests for gateway-level resources are not expected to be highly dynamic to justify the communication costs of modeling the inter-cluster resource allocation as a distributed constraint satisfaction.

Another approach to distributed resource management that handles dynamic changes in resource requirements through the lifetime of a single task is discussed in [16][17]. The idea is to continuously monitor resource usage at each node via a resource manager module. When a node is about to run out of resource capacity risking the fulfillment of task timing constraints, the resource manager of that node will establish negotiations with its counter parts at other nodes to migrate some tasks and free up some capacity of its local resources. Such approach is resource demanding in itself and might be justified in time-critical application where task timeliness is of great importance. In addition, the philosophy of the described approach is to manage system-wide resources only when there is a local resource crisis not at task release time.

Our approach leverages the role of the command in overall network organization. We model the energy, processing and communication resources of the gateway. The command node performs task arbitration among the gateway with consideration the need

of the task and the efficiency of performing such a task while maximizing the life of the gateway.

2. Task Allocation to Gateways

In this section, we cover the mathematical modeling and algorithm for optimal allocation of tasks to gateways in order to extend the lifetime of the system. First we discuss some schedulability issues and the assumptions we made.

2.1 Schedulability Issues

Task allocation among gateway nodes is generally constrained by the accessibility of sensors, the ability of meeting application-related deadlines and the availability of required communication bandwidth. For example if a sensor is reachable only via gateway GI , collection of sensor readings has to be performed by GI . Similarly tasks assigned to a particular gateway have to be schedulable in order to meet the time constraints imposed by the nature of the application. For instance tracking multiple targets that requires on one gateway a computation time that exceed the periodicity of the sensor readings can lead to processing at half of the sampling frequency and thus losing 50% of the accuracy of the reported tracks of some targets.

Since the feasibility of an allocation is constrained by meeting the timing constraints of the allocated tasks, a detailed schedulability analysis is required for every gateway per possible allocation. Precise schedulability analysis is a NP-Hard problem and thus is computationally prohibitive [18]. To simplify the analysis and the actual scheduling of tasks on gateway nodes, we assume that the processing of data lags the collection of data by at least one cycle. Therefore, the target tracking algorithms running in cycle “ $k+1$ ” will be using sensors’ readings collected in cycle “ k ”, as shown in Fig. 2. Such assumption is very practical and ensures source congruency of the sensor data. Since

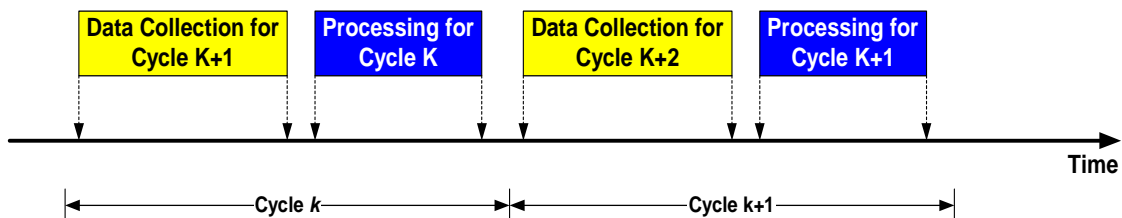


Fig. 2: Processing sensors’ data lags the collection of that data by one cycle

arrival of sensor data is asynchronous (arbitrary) within the cycle, the scheduling of data processing (target tracking) will be extremely unpredictable within the cycle. By giving at least one cycle for collecting the sensor data, we omit dependency between data collection and processing within the cycle and thus ensure data availability when scheduling the data processing task.

2.2 Problem Formulation

We refer to a system-level mission, such as target tracking, as a job, which consists of multiple tasks that perform required communication and processing related to achieving a mission. Tasks of a particular job are assigned to one or multiple gateways. For example, a job for target tracking consists of sensors' data collection, data routing (if needed) and data processing tasks. We formulate the problem of task allocation among gateways as an optimization problem. The objective of the optimization is to maximally extend the life of all gateways, while attaining satisfactory mission results, by balancing the load proportional to the energy that each gateway has. We assume that the relationship between power consumption and both CPU processing and communication is known a priori, so that the energy needed for task execution can be estimated during the allocation. We also assume that any gateway node consumes the same amount of energy to send the same size of data to any other gateway node in the network. In the presented formulation, tasks previously assigned to a particular gateway are not subject to reallocation (migration) to different gateways during allocation of a new set of tasks. We further assume that the communication latency among gateways is bounded and negligible compared to the cycle time. It should be noted that if the communication latency is not negligible compared to the cycle time, additional lag cycles between data collection and processing might have to be enforced in order to ensure timely data availability.

Before describing the objective function and the conditions for the task allocation optimization, we define the following definitions and notations to be used.

- N Number of gateway nodes in the system
- m Number of sensors needed to track a target
- t_o System start time
- t_b Current time before allocation

- t_a Time after allocation
- τ System cycle time, which is determined by the frequency of the arrival of sensor readings (if they are similar), or the least common multiple of the different reading frequencies.
- $\alpha_i(t)$ Available processing capacity per cycle of gateway node i at time t .
- $g\beta_i(t)$ Available inter-gateway communication bandwidth, per cycle for gateway node i at time t .
- $s\beta_i(t)$ Available sensor-gateway communication bandwidth, per cycle for gateway node i at time t .
- Ψ The set of jobs to be allocated to gateways. Each job $J \in \Psi$ is a $(m+1)$ -tuple $(p, c1, c2, c3 \dots cm)$ where p is the computation load, and $c1, c2 \dots cm$ are the communication loads for the m sensors required for the job.
- D_Ψ The duration, in cycles, for which the jobs in Ψ need to be performed.
- Φ The set of tasks to be allocated to gateways. Each job $J_k \in \Psi$ will have $m+1$ tasks in Φ , namely $J_{k,p}, J_{k,c1}, J_{k,c2}, \dots, J_{k,cm}$ corresponding to the computation and communication loads of the job, respectively.
- ϕ_i The set of tasks allocated to a gateway node i , where $\Phi \supseteq \phi_i$.
- $E_i(t)$ Remaining energy at gateway node i at time t .
- $rEU_i(t)$ Rate of energy consumption (usage) per cycle of gateway node i at time t , which is the sum of all energy consumption for computation and communication of all assigned task to gateway node i ($i \leq N$). It should be noted that changes in this rate due the termination of tasks have to be tracked by the command node so that the actual energy consumption rate is referred to during task allocation. The command node can track
- $L_i(t)$ Remaining life in cycles of the gateway i at time t , where $L_i(t) = E_i(t)/rEU_i(t)$.
- $MRL_i(t)$ Minimum remaining life in cycles of the gateway i at time t , where $MRL_i(t) \leq L_i(t)$. It results from previous commitments to perform jobs assigned in previous allocations to the time extend of these jobs.
- $ePEU_i(J_p)$ Estimated energy consumed by computation per cycle to process a task J_p at a gateway node i ($i \leq N$).
- $eCEU_i(J_c)$ Estimated energy consumed per cycle by communication to perform a task J_c at a gateway node i ($i \leq N$), including required computation for data collection. For sensors unreachable by the gateway (outside the cluster) this function equals to ∞ , since it not possible to allocate such task to this gateway.
- $eRPEU_i(J_c)$ Estimated energy consumed per cycle for routing collected data of a task J_c at a gateway node i ($i \leq N$) to another gateway (assuming equal communication energy-usage rate between any two gateways).

- $eP_i(J_p)$ Estimated processing capacity per cycle required for performing the computation for a task J_p at a gateway node i ($i \leq N$).
- $eRP_i(J_c)$ Estimated processing capacity per cycle required for routing the sensor data of a task J at a gateway node i ($i \leq N$) to another gateway (assuming equal processing need for communication between any two gateways).
- $eGC_i(J_c)$ Estimated inter-gateway communication bandwidth per cycle needed by a gateway node i ($i \leq N$) to perform a task J_c .
- $eSC_i(J_c)$ Estimated sensor-gateway communication bandwidth per cycle needed by a gateway node i ($i \leq N$) to perform a task J_c .

If the set of jobs Φ could be successfully allocated to the N gateways, the following the relationships hold:

1. $\Phi = \bigcup_{i=1}^N \varphi_i$ Ensuring that all the tasks are successfully allocated
2. $\bigcup_{i \neq j} \varphi_i \varphi_j = \emptyset \quad \forall i, j \leq N$ Every task is uniquely assigned to one gateway.

Since no resource reclamation is considered while allocating new tasks (previously allocated tasks are not subject to migration to other gateways), the power consumed by the load of a gateway at t_b will not change. Therefore, the rate of energy consumption at gateway node i after the allocation, is the rate of power consumption of the recently allocated tasks added to the consumption's rate prior to the allocation.

$$\begin{aligned}
rEU_i(t_a) = rEU_i(t_b) &+ \sum_{\forall J_{k,p} \in \varphi_i} ePEU_i(J_{k,p}) + \sum_{\forall J_{k,c} \in \varphi_i} eCEU_i(J_{k,c}) + \sum_{\substack{\forall J_{k,c} \in \varphi_i \\ \text{where } J_{k,p} \notin \varphi_i}} eRPEU_i(J_{k,c}) \\
&+ \sum_{\substack{\forall J_{k,c} \in \varphi_i \\ \text{where } J_{k,p} \notin \varphi_i}} eRRecvU_i
\end{aligned}$$

The minimal remaining life of a gateway node i , $MRL_i(t_b)$, reflects the number of cycle for which the node ought to last at the current energy consumption rate in order to perform the jobs it has for the duration required. After successfully allocating new tasks to that gateway, the new minimal remaining life, $MRL_i(t_a)$, is calculated as follows:

$$\begin{aligned}
MRL_i(t_a) &= \frac{\text{Energy required to finish all jobs}}{\text{New rate of energy consumption}} \\
&= \frac{MRL_i(t_b) * rEU_i(t_b) + D_\Psi * [rEU_i(t_a) - rEU_i(t_b)]}{rEU_i(t_a)}
\end{aligned}$$

$MRL_i(t_a)$ simply indicates the number of cycle the gateway energy would last performing the jobs at the new energy consumption rate $rEU_i(t_a)$. It should be noted that the new minimal remaining life of the gateway could not exceed the longer of old and new allocated jobs, i.e. $MRL_i(t_b) \leq MRL_i(t_a) \leq \max(D_\Psi, MRL_i(t_b))$, since for $D_\Psi \neq MRL_i(t_b)$ the new rate of power consumption might be dropped after a number of cycles equals to $\min(D_\Psi, MRL_i(t_b))$.

Objective Function: The objective of the allocation is to maximize the life of the clusters by maximizing the sum of weighted-value of the remaining life of all gateways. A High weight reflects the importance of extending the life of a particular gateway for current and future missions.

$$\text{Max} \sum_{i=1}^N W_i * L_i = \text{Max} \sum_{i=1}^N W_i * \frac{E_i(t_b)}{rEU_i(t_a)} \equiv \text{Min} \sum_{i=1}^N W_i * \frac{rEU_i(t_a)}{E_i(t_b)}$$

Constraints: The task allocation optimization is subject to the following conditions:

1. **Schedulability condition:** The allocated tasks have to be schedulable at the gateway nodes so that each task will meet the timing constraints. By assuming at least one cycle of time lag between data collection and processing, dependencies among tasks running within the same cycle are removed. Therefore the allocated tasks are schedulable if the gateway has sufficient processing capacity to execute all of these tasks.

$$\alpha_i(t_b) \geq \sum_{\forall J_{k,p} \in \varphi_i} eP_i(J_{k,p}) + \sum_{\substack{\forall J_{k,c} \in \varphi_i \\ \text{where } J_{k,p} \notin \varphi_i}} eRP_i(J_{k,c}) \quad \forall i \leq N, J_k \in \Psi$$

2. **Bound on the available communication bandwidth per gateway for interaction with sensors in the cluster:**

$$s\beta_i(t_b) \geq \sum_{\forall J_{k,c} \in \varphi_i} eSC_i(J_{k,c}) \quad \forall i \leq N, J_k \in \Psi$$

3. **Bound on the available communication bandwidth per gateway for interaction with other gateways:**

$$g\beta_i(t_b) \geq \sum_{\substack{\forall J_{k,c} \in \varphi_i \\ \text{where } J_{k,p} \notin \varphi_i}} eGC_i(J_{k,c}) \quad \forall i \leq N, J_k \in \Psi$$

- The gateway needs to have available enough power to perform the newly allocated tasks for the required duration D_Ψ , in addition to the energy needed to fulfill the previous commitment for existing tasks for MRL_i .

$$MRL_i(t_b) * rEU_i(t_b) + D_\Psi * [rEU_i(t_a) - rEU_i(t_b)] \leq E_i(t_b) \quad \forall i \leq N$$

2.3 Optimization Algorithm

The complexity of a general optimization problem depends on the linearity of the objective function and the constraints [19]. Our model can be classified as a zero-one nonlinear optimization problem, for which the technique of simulated annealing proved to be very effective [20][21]. The Simulated Annealing Optimization method distinguishes between different local optima and has been successfully applied to similar problems [22]. The validation experiments, as described next, used an implementation of the simulated annealing algorithm developed by Taygeta Scientific Inc. [23].

3. Experiments and Analysis

Recall that gateways' remaining energy is at the core of the objective function. For experimental purposes, we formulated variants of the objective function based on three metrics. The metrics that we considered and corresponding objective functions are shown in Table 1.

Table 1: Metrics and corresponding Objective Functions

Metric	Objective Function
Minimum Remaining Energy	$Max(\underset{i}{Min}(W_i * L_i))$
Total Remaining Energy	$Max\left(\sum_{i=1}^N (W_i * L_i)\right)$
Average Remaining Energy	$Max(\underset{i}{Avg}(W_i * L_i))$

3.1 Experimental Setup

The parameters chosen for the experiments are listed in Table 2. In order to predict the energy consumption for computation and communication for a gateway node, we utilize the energy models mentioned in [5]. In that work a computation task involve the execution of a set of the algorithms like fast Fourier transform. In the experiment, the energy required for performing a computation task is computed as a random summation of one or more of the energy dissipation values required by the mentioned algorithms.

The communication model for energy dissipation assume that the radio dissipates $E_{elec} = 50$ nJ/bit in the transmitter or receiver circuitry and $\epsilon_{amp} = 100$ pJ/bit/m² for the transmit amplifier. An r^2 energy loss due to channel transmission is assumed. Thus, the transmission E_{Tx} and receiving E_{Rx} energy for a k -bit message a distance d , are:

$$E_{Tx}(k, d) = E_{Tx-elec}(k) + E_{Tx-amp}(k, d) = E_{elec} * k + \epsilon_{amp} * k * d^2$$

$$E_{Rx}(k) = E_{Rx-elec}(k) = E_{elec} * k$$

Based on the above model, we calculated the energy required by a gateway to route data to another gateway. The distance between the two gateways is considered to be between 2 and 20 km. Sensors are picked for the generated tasks using a normal distribution.

Table 2: Parameters for the experiment

Total number of gateways	5
Load (the total number of jobs to be performed)	10 to 200
Total number of communication tasks corresponding to a job j	3
Remaining energy at a gateway node i (in joules)	5
Weight associated with gateway	1
Rate of energy consumption at a gateway before new task allocation (in joules)	0.000123
Energy required for performing a communication task by a gateway (in joules)	0.000005
Energy required for FFT (in joules)	0.005389
Energy required for DCT (in joules)	0.0001
Energy required for IDCT (in joules)	0.00013
Energy required for FIR (in joules)	0.00123
Energy required for TDLMS (in joules)	0.002129
Energy required by a gateway i for receiving data from any gateway (in joules)	0.00000005

3.2 Results and Analysis

After the optimal allocation of tasks to the gateways, we calculated the remaining energy for each gateway by using the objective functions mentioned in section 3.1. We tried to maximize the minimum, total and average remaining energy of all the gateways and obtained the results depicted in figures 3, 4 and 5 respectively. Each graph shows the results of one particular objective function generated by the optimization algorithm and also the results obtained without performing any optimization. The baseline allocation, with no optimization, assigns a task to the gateway whose cluster includes the most number of sensors involved in that job.

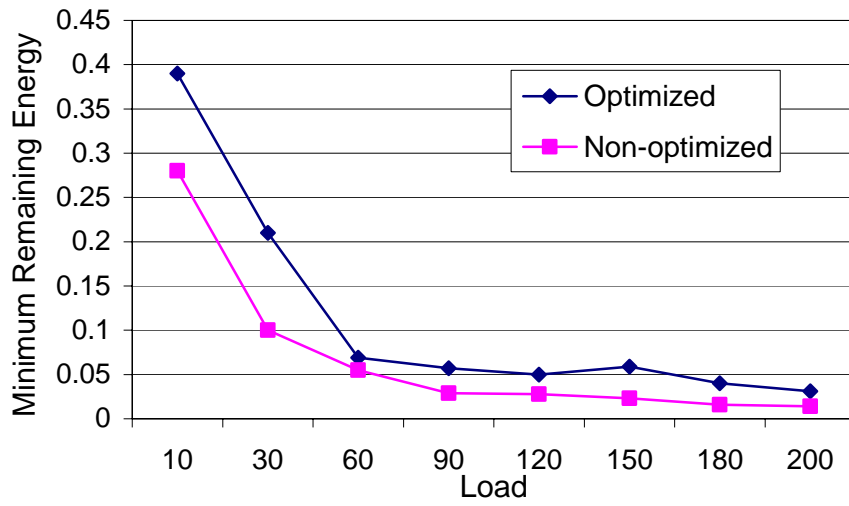


Fig. 3: Minimum remaining energy of gateways for different loads

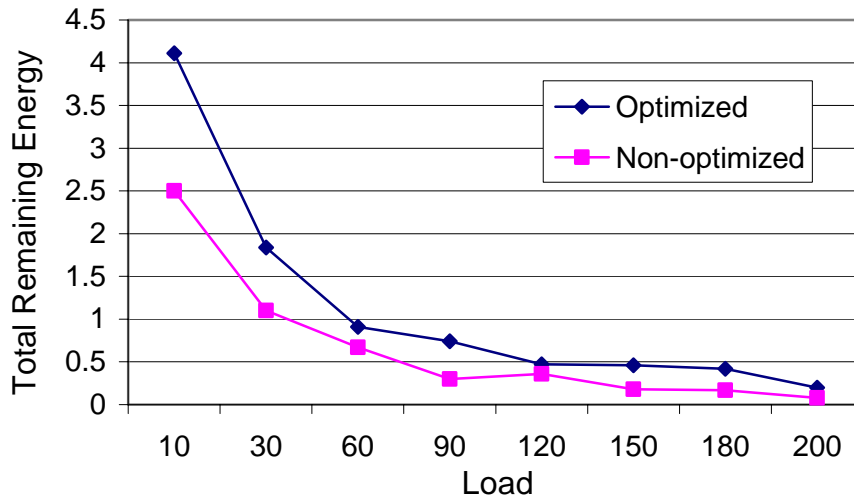


Fig. 4: Total remaining energy of gateways for different loads

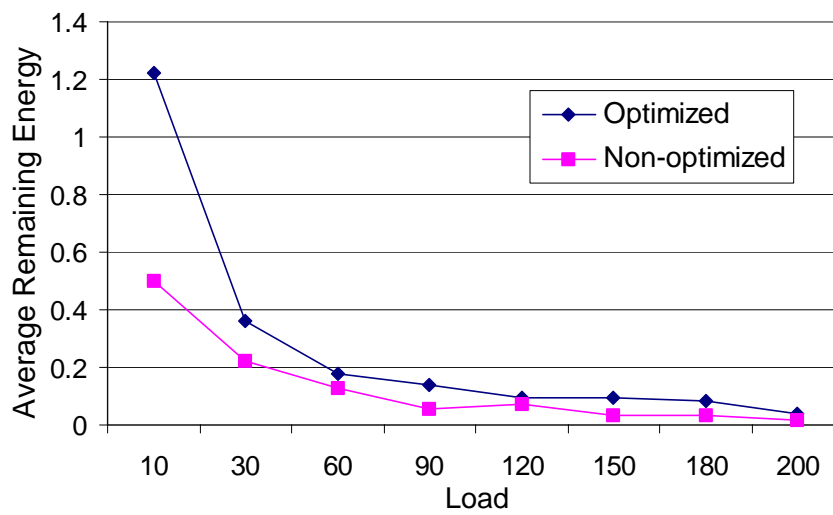


Fig. 5: Average remaining energy of gateways for different loads

The results show that that our optimized task allocation lowers the rate of energy consumption at the gateways. It is worth noting that the descent in the energy curves for all metrics are mainly due to the increased use of gateway energy for larger number of tasks. In case of maximizing the minimum gateway energy as shown in Fig. 3, our approach almost doubles the minimal remaining energy at the end of task execution. This metric also can lead load balancing for scenarios, in which one cluster is overwhelmed with tasks. Monitoring the minimal energy reserve at all gateways, allow tasks to be assigned to lightly loaded clusters.

For the total remaining energy metric, Fig. 4, some performance gain could be achieved. Such performance gain is only due to our optimal handling of tasks that have a maximum of one sensor per cluster. For these tasks, the gateway of the cluster that includes the first sensor on the list will be picked in the baseline allocation. The optimization in this case minimizes the energy for inter-gateway communication. However, at high load the frequency of this scenario diminishes and the performance of the optimized allocation becomes close to the baseline. It is worth noting that we run the experiment using uniform distribution for sensor selection and found that the performance of the optimized approach is consistently better than the baseline allocation using this metric. Since the sensor selection for a task is application-dependant, we envision total remaining energy metric to be useful for applications that require reading from distant locations, e.g. forest fire monitoring and detection of harsh weather phenomena. Similar observations could be made for the average remaining energy metric, depicted in Fig. 5.

4. Conclusion and Future Work

In this paper, we described a method for optimal allocation of tasks to gateways in a cluster-based sensor network. The presented approach maximizes the lifetime for the gateways and eventually for the whole network. The problem of allocation of tasks is modeled as a non-linear constrained satisfaction problem and implemented by using simulated annealing optimization technique. Three different objective functions, aiming at the minimum, average and total remaining energy of gateways are considered. Simulation results have shown that optimized allocation of tasks always give better

results than non-optimized case in terms of energy usage. In addition, the results indicated that maximizing the minimum gateway energy fits well with application in which some clusters are used more than the others. On the other hand maximizing total or average remaining gateway energy can suit applications that require reading from distant locations.

Our future plan includes extending the system architecture to support multi-command node and multi-gateway. The presence of multi-command node would require the task allocation to be performed in a distributed manner. To avoid conflicting strategy among the command nodes, task arbitration has to be performed at the gateway in such setup. We are planning to investigate lightweight distributed optimization mechanisms that fit such dynamic and resource-constrained network architecture.

References

- [1] D. Estrin and R. Govindan, "Next century challenges: Scalable coordination in sensor networks," *Proc. Mobicom, 1999*, pp. 263-270.
- [2] L. Subramanian and R. H. Katz, "An Architecture for Building Self Configurable Systems," *IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing*, August 2000.
- [3] K. Sohrabi, J. Gao, V. Ailawadhi, G. J. Potie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, October 2000, pp. 16-27.
- [4] M. Younis, M. Youssef, K. Arisha, "Energy-Aware Routing in Cluster-Based Sensor Networks," *Proceedings of the 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS2002)*, Fort Worth, Texas, October 2002.
- [5] Wendi Rabiner Heinzelman, Amit Sinha, Alice Wang, and Anantha P. Chandrakasan, "Energy Scalable Algorithms and Protocols for Wireless Microsensor Networks," *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP '00)*, June 2000.
- [6] Buczak and V. Jamalabad, "Self-organization of a Heterogeneous Sensor Network by Genetic Algorithms," *Intelligent Engineering Systems Through Artificial Neural Networks*, C.H. Dagli, et. a.. (eds.), Vol. 8, pp. 259-264, ASME Press, New York, 1998.
- [7] Chunhung Richard Lin and Mario Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, Vol. 15, No. 7, September 1997.
- [8] E. Huh, et al., "Accommodating QoS prediction in an adaptive resource management framework," in *Parallel and Distributed Processing*, J. Rolim et al. eds., *Lecture Notes in Computer Science*, Vol. 1800, pp. 792-799, Springer-Verlag, New York, NY, 2000.
- [9] D. Fernandez-Baca, "Allocating modules to processors in a distributed system," *IEEE Transaction on Software Engineering*, Vol. SE-15, pp. 1427-1436, November 1989.

- [10] *Integer and Combinatorial Optimization*. G. Nemhauser and L. Wolsey, Wiley-Interscience, New York, 1999.
- [11] S. Ali, et al. "Greedy Heuristics for Resource Allocation in Dynamic Distributed Real-Time Heterogeneous Computing Systems," in the *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'02)*, Las Vegas, Nevada, June 2002.
- [12] M. Maheswaran, et al., "Dynamic mapping of a class of independent tasks onto heterogeneous computing systems," *Journal of Parallel and Distributed Computing*, Vol. 59, pp. 107–131, Nov. 1999.
- [13] H. Siegel and S. Ali, "Techniques for Mapping Tasks to Machines in Heterogeneous Computing Systems," *Journal of Systems Architecture, Special Issue on Heterogeneous Distributed and Parallel Architectures: Hardware, Software and Design Tools*, 2000.
- [14] B. Krishnamachari, R. Bejar, and S. Wicker, "Distributed Constraint Satisfaction and the Bounds on Resource Allocation in Wireless Networks," *Proceedings of the International Symposium on Communications Theory and Applications*, July 2001.
- [15] P. Modi, et al., "Dynamic Distributed Resource Allocation: A Distributed Constraint Satisfaction Approach," *Proceedings of the International workshop on Agent Theories, Architectures, and Languages (ATAL'01)*, 2001.
- [16] D. Ivan Rosu, K. Schwan, S. Yalamanchili, and R. Jha, "On adaptive resource allocation for complex real-time applications", *Proceedings of the 18th IEEE Real-Time Systems Symposium*, San Francisco, December 1997.
- [17] Binoy Ravindran, Lonnie R. Welch and Behrooz A. Shirazi, "Resource Management Middleware for Dynamic, Dependable Real-Time Systems," *The Journal of Real-Time Systems*, Vol. 20, pp. 183-196, 2001.
- [18] W. Halang and A. Stoyenko, *Constructing Predictable Real-Time Systems*. Boston-Dordrecht-London: Kluwer Academic Publishers, 1991.
- [19] P. Gray, et al., "A Survey of Global Optimization Methods", *Technical Report*, Sandia National Laboratories, Albuquerque, New Mexico, 1997.
- [20] Optimization by Simulated Annealing, *Science*, V. 220, No. 4598, pp. 671 – 680; Kirkpatrick, S., C.D. Gelatt Jr, and M.P. Vecchi, 1983.
- [21] R. Diekmann, R. Lüling, J. Simon, "A General Purpose Distributed Implementation of Simulated Annealing," in the *Proceedings of the 4th IEEE Symposium on Parallel and Distributed Processing (SPDP'92)*, 1992.
- [22] Marco DiNatale, John A. Stankovic, "Applicability of Simulated Annealing Methods to Real-Time Scheduling and Jitter Control," in the *Proceedings of the 16th IEEE Real-Time Systems Symposium (RTSS'95)*, Pisa, Italy, 1995.
- [23] <http://www.taygeta.com/annealing/simanneal.html>