DYNAMIC TASK ALLOCATION IN MULTI-HOP WIRELESS SENSOR NETWORK WITH MOBILITY

Sri Vaishnavi R[1], Sukumar M[2]

[1] PG scholar,Department of computer science and engineering , Sri Vidya College Of Engineering & Technology, Tamilnadu, India

[2] Assistant professor, Department Of Information Technology, , Sri Vidya College Of Engineering & Technology , Tamilnadu, India

ABSTRACT

In wireless sensor network (WSN), the resources usage is highly related to the tasks execution which consume a positive amount of computing and communication bandwidth.This Paper Presents a task allocation oriented framework to enable an efficient in-network processing and cost-effective resource sharing for dynamic multi-hop wireless sensor networks with low node mobility, e.g., pedestrian speeds. The proposed system incorporates a fast task reallocation algorithm to quickly recover from possible network service disruptions, such as node or link failures.An evolutional self-learning mechanism based on a genetic algorithm continuously adapts the system parameters in order to meet the desired application delay requirements, while also achieving a sufficiently long network lifetime. Since the algorithm considers the time delay while updating task assignment, to introduce an adaptive window size to limit the time delay periods and ensure an up-to-date solution based on node mobility pattern and device processing capabilities. The results show considerable performance improvement in extending network lifetime. Furthermore, the proposed framework will provide noticeable reduction in the frequency of missing application deadlines

Keyword : - *Minimum hop count,Time delay , Fast Task Recovery Algorithm ,Task Allocation.*

1. INTRODUCTION

In WSN ,the task allocation process is needed to support high performance application in multi-hop multimedia wireless sensor networks (MWSNs) with limited node capabilities for resource sharing and node collaboration [1]. Each sensor node has certain capabilities that are computation capacity , power supply and communication ability. In WSN , the resource usage is highly related to the executing of tasks which consume the bandwidth of computing and communication [2]. In-network processing for wireless sensor network (WSN) has more energy-efficient than sending all the raw data to the end user. It improves the real-time performance and also reduce the communication volume of WSN. The Nodes cannot only act as independent processing elements in WSN. It is capable to collaborate with each other during a direct or multi-hop communication links. So, the network can be taken as a parallel computing system for essential data exchange among nodes. The processing task can be divided into smaller sub tasks and then executed at the same time on independent sensor nodes. The execution of each subtask for every node would consider a certain amount of resources for computing and communication. An efficient mechanism is required to allocate each task to appropriate group of nodes while meeting the application requirements[3].The sensor nodes cannot be finished a complex task in a particular time due to their limited resource and processing capacity. If one sensor node runs out of its energy, the task allocation algorithm needs to run the entire steps from beginnings,which wastes large amounts of energy and cannot satisfy the real-time requirement of the application[4].

Tasks would be allocated to sensor nodes with the concern of energy conservation and balanced energy consumption to elongate the network lifetime. In the meantime, many applications involve that the tasks should be finished in a short time. So, the task execution time should also be considered an optimization metric.The selected cluster nodes focused on task ,that must have an adequate resource and fitness for finishing the essential workload. The selected nodes are able to connected with each other through direct or multi-hop links for essential data exchanges.

In task allocation, the problem is how to assign a task to its most suitable sens or node and concurrently balance the network load in the context of the tentative and dynamic network environments denotes an important and critical issue in WSN.In sometimes, the node leaves the network, due to communication problem or physical node failu re, serious consequences, such as network service disruption, can occur. In such cases, control messages are exchanged among nodes in order to isolate the faulty ones and detect the affected tasks that need to be immediately reallocated to suitable nodes.The task workload and the connectivity are considered as constraints to ensure the accomplishment of each task and essential data exchange among selected nodes. A hybrid fitness function involving task execution time, energyconsumption and network lifetime is designed to evaluate the quality of each particle.

The rest of this paper is organized as follows. Related work is reviewed in section II. The Preliminaries in section III. Task Allocation and Scheduling in MWSNs is proposed in section IV. The DTAS Framework described in section V. Performance in section VI. Conclusions are given in section VII.

2. RELATED WORK

The task allocation problem in parallel and distributed systems has been extensively studied in both wired and wireless networks. Existing solutions are based on multi-objective optimization approaches were considered: minimizing task completion time [15], reducing energy consumption [8], load balancing to achieve an equalized node lifetime and maximizing service reliability [2]. In wired networks, since nodes are often connected with dedicated and high quality links, communication costs and delays are often considered to be negligible.

However, the situation in an MWSN is quite different, and solutions like [7,9] and consider both processing costs and wireless communication costs.processing high level tasksmay still exceed the capacity of powerful nodes. Moreover, the energy balancing issues are not discussed explicitly. To extend the network lifetime of WSN, some solutions than focus on the energy balancing issues [14], For instance, in order toachieve balanced energy consumption and extended network lifetime when distributing in-network processing task to WSN.The authors of [17] proposed an adaptive intelligent task mapping and scheduling scheme using GA, where a hybrid fitness function is designed to balance the workload of network nodes with guaranteed application deadlines. Moreover, to achieve better performance, more intelligentoptimization technology should be explored for the task allocation problem of WSN. BPSO is the binary version of PSO and has the potential to solve many binary optimization problems of WSN.

In this paper, the Dynamic Task Allocation and Scheduling (DTAS) framework is presented. DTAS aims at minimizing the frequency of instances when an application misses an arbitrarily set deadline (*deadline misses ratio*), while also extending the network lifetime by balancing node energy consumption levels. DTAS can be summarized as follows: First, a heuristic minimum hop count algorithm is designed to guide the initial solution creation, which can effectively reduce problem complexity. Second, a self-learning process (SLP) based on a GA is applied, which continuously evolves a set of solutions, so that multiple design objectives can be met.The fitness function in SLP initially favors meeting the deadline requirement and, then, gradually leans towards a balanced solution between task execution time and network lifetime.Finally, to deal with sudden node or link failure events and to update the solutions in SLP, a Fast Task Recovery Algorithm (FTRA) is designed to quickly reallocate faulty task assignments.

3. Preliminaries

3.1. System Models

A Directed Acyclic Graph (DAG) G = (T,E) is used to model an application [11]. Each vertex in the DAG represents a task $T_i \in T$ that is connected to other vertices by directed edges. Each task, T_i has a workload, p_i representing the processing requirement in terms of the number of CPU clock cycles to execute the task.The weight of each edge, e_{ij} stands for the amount of data transmitted from T_i to T_j .A direct edge($e_{ij} \in E$)shows the precedence relations among tasks, i.e. T_i should be completed before T_j.Therefore, a DAG has a topological task execution order, which we term the task scheduling sequence (TSS). Furthermore, an application can iteratively execute the DAG. A round is defined as the time period of a DAG execution cycle.

The network topology consists of a total number of M heterogeneous nodes $V = \{v_1, v_2, \cdots v_M\}$ that are randomly deployed in the network. For simplicity, transmission power control is not enabled. Hence, all nodes have a fixed communication range, and they are connected via multi-hop links. Nodes are battery powered, and each node has a finite energy supply that is not refilled. Heterogeneous initial battery energy and processing speeds are considered. A non-preemptive scheduling policy is adopted, so that only one job can be processed at each node at a time. It is assumed that nodes are synchronized and that the wireless channel condition is stable. Furthermore, in order to perform scheduled multi-hop communication, a bandwidth reservation mechanism is used, such as a TDMA (time division multiple access) based MAC (media access control) protocol .Unless specified otherwise, each task is executable at every node.However, each node has regular chat message exchanges with its neighbors and periodically reports its own neighbor list to a central network controller (the gateway). Based on the collected information, the network link topology, L, is updated periodically. A dedicated control channel is used for these message exchanges, whose energy consumption is included in the total cost calculation.

3.2. Definitions

The terms used in the rest of the paper are as follows:

1. *Network lifetime (NL):* The time period until the first node fails due to energy depletion.
2. *Schedule length (SL):* The execution time of a DAG.

3.3. Problem Definition

The problem that this paper addresses is two-fold. First, an optimized task allocation solution *s*, isto be found with the objective of maximizing the network lifetime, NL, under the required time-delay constraints. To achieve this, the total schedule length,SL, must meet the deadline,$t_{deadline}$. Hence, the objective function can be formulated as follows

$$\max\{NL(s),\ \ s \in \text{total search space}\}$$
$$\text{subject to} : SL(s) \leq t_{deadline} \quad (1)$$

Secondly, the chosen solution, s, should be able to update itself, such that it can adapt to networkdynamics. However, this is a challenging task because of the following reasons:

1. *Node mobility and node failure events:* The optimized task allocation solution may become invalid when such events occur. Re-assigning the affected tasks can only serve as a temporary solution, as re-optimization is required according to emerging network conditions. Nevertheless, due to the problem complexity, a complete re-run of the algorithm is costly.
2. *Algorithm runtime and complexity:* The proposed task allocation algorithm runs on the gateway node, and its algorithm runtime is denoted by K. In static networks, a high-cost algorithm canwork perfectly well as an off-line solution. On the other hand, algorithm runtime is critical indynamic environments. Since optimization parameters have to be quickly modified in order to adapt to changing conditions, optimization procedures that require a large value of K to complete are likely toproduce outdated solutions in dynamic environments.

4. Task Allocation and Scheduling in MWSNs

In a DAG, G, a task pair (T_i, T_j) connected by a directed edge, e_{ij} , could be allocated to nodesthat are several hops away from each other in the network. Therefore, multi-hop communication costs must be included in the task allocation solution structure. Furthermore, task scheduling in an MWSN needs to take into account particular issues, like parallel processing among independent nodes, possible simultaneous communications and multi-cast transmissions. To tackle these issues, in our previous work [13], we developed a task allocation model and a multi-hop scheduling mechanism for static MWSNs. Since the proposed DTAS presented in Section 4 is based on this model, we briefly describe it in this section.

4.1. Multi-Hop Extension of Task Allocation

For a solution, s, to be evaluated in a multi-processor environment, first, an encoding processtransforms s into individual tasks that can be independently processed.It contains a mapping of a three-task DAG to a four-node network. The elements in the first row are the tasks, while the corresponding places in the second row and third row stand for node ID and computation load, respectively. By observing either the matrix C or the network, it can be seen that T_1 is allocated to v_1, and T_1's child tasks, T_2 and T_3, are allocated to v_3 and v_4, respectively. Figure 2a also demonstrates the communication relation amongst tasks, modeled by a three-by-γ matrix, *E*, called the *edge*, where is equal to the total number of edges in the DAG. The three elements in each column of *E* represent the sender task (T_1), the receiver tasks (T_2 or T_3) andthe total amount of data (e_{12} or e_{24}) that need to be transmitted.

4.2.Computation of the Network Lifetime (NL)

In order to calculate the expected lifetime NL(s), the computational costs, pi, and the edge costs, eij ,first need to be converted into the actual time and energy costs at the assigned nodes, based on processing speeds and communication distances. NL(s) is calculated by:

$$NL(s) = \min\{ \frac{R^{v3}}{E^{v3}{}_{total}} \mid j = 1, 2, \cdots, M\} \qquad (2)$$

where Rv denotes node v's residual energy level and Ev total is the total energy consumption during one round of DAG execution at node v. Rv can be obtained from periodic node reports whose signalling co st is explained.

4.3. Computation of the Schedule Length (SL)

Based on HC and HE, multi-hop scheduling should provide a suitable schedule length, SL, thatenables simultaneously occurring communication and parallel processing events. However, interferencebetween different transmission events and the overlap of task execution at each node should be avoided. Therefore, the same scheduling method proposed in [13] is applied, where a two-hop interference model [9] is used and a medium access delay is introduced, such that the sender of a communication event does not cause interference on its one-hop receivers, and *vice versa*. Details of computation scheduling and communication tasks can be found in [13].

5. . The DTAS Framework

The task allocation in multi-hop wireless networks shown in the previous section is already acomplex process and has been shown to be NP-hard (Non-deterministic Polynomial-time hard) [11],while network dynamics further complicates the problem. For instance, node mobility and failure events can easily render a task allocation solution invalid, in which case, a complete re-run of the task allocation algorithm from scratch is not a feasible option, since this is computationally inefficient.

DTAS has the following three main components:

1. *Self-learning process (SLP)*: SLP is a periodically operated GA-based system component that runs in the system background and performs parallel optimization of task allocation solutions. Unlike conventional GAs, solutions at each evolutional stage of SLP can be modified based on changes in network topology. Hence, SLP results can be continuously updated and evolved.

2. *Fast Task Recovery Algorithm (FTRA)*: FTRA is a low-complexity event-triggered system component, which updates SLP solutions. FTRA can quickly perform task re-allocation when node or link failures occur.

3. *Task Re-allocation Decision Maker (TRDM)*: TRDM interacts with other system components and makes task re-allocation decisions based on different network conditions.

5.1. The Minimum Hop Count Algorithm (MHC)

MHC is used for system initialization, as well as being implemented in the FTRA algorithm toreallocate tasks when network failure events occur. Since a fast system response is normally expected for these two processes,MHC is designed to assign tasks based on hop distance only, rather than calculating SL and NL. This is because hop distance directly affects communication costs, which normally dominate the total consumption (in both time and energy) [13,9]. Therefore, a hop distance-based fuzzy search can efficiently reduce algorithm execution time and provide quick sub-optimal solutions to the system.

Algorithm :1The Minimum Hop Count (MHC) algorithm.

1: At node vi:
2: **for** each T ∈ G based on a task scheduling sequence (TSS) **do**
3: *candidates*← ∅
4: **if** Tpre = ∅**then**
5: Assign T as a source task to vi;
6: continue;
7: **else**
8: Determine Vpre
9: **for** each node vi∈ V **do**
10: THCi ← 0;
11: **for** each vj∈Vpre**do**
12: THCi ← THCi + HCvjvi ;
13: end for
14: end for
15: end if
16: for each node vi∈ V do
17: if THCi ≤ min(THC) + n
then
18: candidates ← {candidates, vi};
19: end if
20: end for
21: % randomly select a node from *candidates*
22: Node(T) = rand(candidates);
23: end for

5.2 The Fast Task Recovery Algorithm (FTRA)

When an active node failure (Vf), link failure (Lf) or multiple simultaneous failure events take place,event-triggered reports containing information about those failure events and corresponding network topology changes are sent back to the gateway (Please note, not all node/link failure events would effect the current allocation s_, which are not belonging to Vf and Vl, e.g., a node fails, but with no tasks assigned.). The FTRA algorithm is then used to perform task re-allocations. The FTRA algorithm is shown in Algorithm 2. When a node, vi, in C fails (line 3), its tasks have to be re-allocated. If any T ∈Tdefect is a source task (line 8), then FTRA randomly assigns this task toone of the neighbour nodes. Otherwise, the MHC algorithm (line 13) is used to choose the replacement node. Then, multi-hop extension is performed (line 21) in order to avoid any resulting broken links.

Algorithm2:The Fast Task Recovery Algorithm (FTRA) algorithm.

1: % Detect the set of defected tasks T defect
2: for each node vi∈C do
3: if vi∈vf then
4: Include all T assigned on vi in Tdefect
5: % Fix node failure
6: for each T ∈Tdefect do
7: Find all Vpre for T
8: if Vpre = ∅ then
9: % Re-allocate source tasks
10: ni ← vi's one-hop neighbours
11: % Randomly select a node from ni
12: Node(T) = rand(ni)
13: else

14: Node(T) = MHC(vi)
15: end if
16: Update C with T and Node(T)
17: end for
18: end if
19: end for
20: % Fix possible link failure: perform multi-hop extension
21: C =⇒ HC, E =⇒ HE

6. Conclusion

In this paper, the DTAS framework is proposed for multi-hop multimedia wireless sensor networkswith lowmobility nodes, which can minimize the deadlinemiss ratio while also preserving and balancingnode energy levels to extend network lifetime. This task allocation problem is very challenging when network dynamic and multi-hop wireless communication aspects are addressed simultaneously. A fast, but simple, heuristic algorithm, like Greedy, may only provide sub-optimal solutions. On the other hand, a sophisticated heuristic search algorithm, like MTMS, or a conventional GA-based solution, suchas ITAS, performs relatively well under static network conditions, but has poor adaption to network dynamics, due to algorithm time-complexity. An integration of such a stage GA-based evolutionalalgorithm with an efficient fast heuristic running in between to adjust and correct the GA population is shown to be suitable for solving such complex and dynamic task allocation problems under a slowly changing environment. Furthermore, DTAS is able to make trade-offs between algorithm runtime and performance. Adaptive solutions can be produced according to how fast network changes occur, while also considering the processing capability of a controller device that needs to deal with such changes.

7.References

1.Jun Yang, Hesheng Zhang"Task Allocation for Wireless Sensor Network Using Modified Binary Particle Swarm Optimization", *IEEE SENSORS JOURNAL,VOL.14, NO. 3*, MARCH 2014

2.Y. Tian and E. Ekici, "Cross-layer collaborative in-network processing in multihop wireless sensor networks," *IEEETrans.MobileComput.*,vol.6,no3,pp297–310,Mar2012

3.M. Karakaya and H. Qi, "Distributed target localization using a progressive certainty map in visual sensor networks," *Ad Hoc Netw.*, vol. 9, no. 4, pp. Jun. 2011.

4.Y. Fu, Q. Ling, and Z. Tian, "Distributed sensor allocation for multitarget tracking in wireless sensor networks," *IEEE Trans. Aerosp.Electron. Syst.*, vol. 48, no. 4, pp. 2012

5.E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network aggregation techniques for wireless sensor networks: A survey," *IEEE WirelessCommun. Mag.*, vol. 14, no. 2, pp. 70–87, Apr. 2010.

6.Feng Wang,1,2 Guangjie Han,1,3 Jinfang Jiang,1 Wei Li,1 and Lei Shu3," A Task Allocation Algorithm Based on Score Incentive Mechanism for Wireless Sensor Networks .

7.M. Bal, W. Shen, and H. Ghenniwa, "Collaborative signal and information processing in wireless sensor networks: A review," in *Proc. IEEEInt. Conf. SMC*, Oct. 2013, pp. 3151–3156.

8.H. Wang, N. Agoulmine, M. Ma, and Y. Jin, "Network lifetime optimization in wireless sensor networks," *IEEE J. Sel. Areas Commun.*,vol.28, no.7,pp.1127–1137,Sep. 2010.

9. S. E. Yoo, P. K. Chong, D. Kim, Y. Doh, M. L. Pham, E. Choi, *et al.*, "Guaranteeing real-time services for industrial wireless sensor networks with IEEE 802.15.4," *IEEE Trans. Ind. Electron.*, vol. 57, no. 11, pp. 3868–3876, Nov. 2010.

10.W. Guo, N. Xiong, H. C. Chao, S. Hussain, and G. Chen, "Design and analysis of self-adapted task scheduling strategies in wireless sensor networks," *Sensors*, vol. 11, no.7, pp. 6533–6554, 2011.