



HAL
open science

Joint duty cycle Scheduling, resource Allocation and multi-constrained QoS Routing algorithm

Jamila Ben Slimane, Ye-Qiong Song, Anis Koubâa, Mounir Frikha

► **To cite this version:**

Jamila Ben Slimane, Ye-Qiong Song, Anis Koubâa, Mounir Frikha. Joint duty cycle Scheduling, resource Allocation and multi-constrained QoS Routing algorithm. Ad-hoc, Mobile, and Wireless Networks - 10th International Conference, ADHOC-NOW 2011, Jul 2011, Paderborn, Germany. pp.29-43, 10.1007/978-3-642-22450-8_3. hal-00645479

HAL Id: hal-00645479

<https://hal.science/hal-00645479v1>

Submitted on 29 Nov 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Joint duty cycle Scheduling, resource Allocation and multi-constrained QoS Routing algorithm

Jamila Ben Slimane^{1,2}, Ye-Qiong Song², Anis Koubaa^{3,4} and Mounir Frikha¹

¹ MEDIATRON, Higher School of Communication of Tunis, Tunisia

² LORIA and INPL, Campus Scientifique, BP 239 54506, France

³ CISTER Research Unit, Polytechnic Institute of Porto, (ISEP-IPP), Portugal

⁴ COINS Research Group, Al-Imam Mohamed bin Saud University, Saudi Arabia

Abstract. Wireless mesh sensor networks (WMSNs) have recently gained a lot of interest due to their communication capability to support various applications with different Quality of Service (QoS) requirements. The most challenging issue is providing a tradeoff between the resource efficiency and the multi-constrained QoS support. For this purpose, we propose a cross-layer algorithm JSAR (Joint duty cycle Scheduling, resource Allocation and multi-constrained QoS Routing algorithm) for WMSNs on based multi-channel multi-time slot MAC. To our best knowledge, JSAR is the first algorithm which simultaneously combines, a duty cycle scheduling scheme for energy saving, a resource allocation scheme for efficient use of frequency channels and time slots, and an heuristic for multi-constrained routing protocol. The performance of JSAR has been evaluated, showing that it is suitable for on-line implementation.

1 Introduction

Wireless mesh sensor networks (WMSNs) are expected to support various applications with different QoS requirements. According to novel application requirements, QoS constraints become more and more critical in terms of end-to-end delay, data throughput and packet-error-rate. Also, due to energetic constraints at node level, energy saving remains the most challenging issue. Various cross-layer designs are applicable to WMSNs [2–5], but the majority of proposed approaches generally focus on WMSNs based single-channel MAC protocol and only [5] focus on Multi-channel based WMSNs. However dealing with all the WMSN's requirements requires tight collaboration of all network's layers. Although several cross-layer designs have been proposed for WMSNs to ensure energy efficiency or to improve the network's performance or to support QoS guaranties, few approaches have taken into account simultaneously multiple WMSN's requirements. The authors in [2] propose a cross-layer strategy that explores the trade-off between energy efficiency and packet timeliness in time division multiple access (TDMA) based WSNs, by transmission power allocation and routing path selection schemes. The idea in [3] is based on a traffic balancing inside the network and a judiciously allocation of the retry limit to each link. [4] proposes a cross-layer design based on power transmission management, routing, and duty-cycle schedule to optimize the WSNs energy-efficiency. We find that only [5] proposes a cross-layer design for WMSNs based on multi-channel access. The goal of the joint wakeup/sleep scheduling and

routing algorithm proposed in [5] is the minimization of the communication latency while providing energy efficiency for nodes in FDMA-based multi-channel WMSNs. The proposed routing algorithm takes into account only delay constraint to minimize communication latency and save energy. We note that the flow differentiation, the link reliability, the residual energy per node and the energy consumption per path have not been taken into account during routing process. None of these approaches combines simultaneously multi-constrained QoS routing, duty cycle scheduling and efficient frequency channels and time slots allocation.

The main problem, we address, is designing an efficient cross-layer algorithm that simultaneously takes into account various WMSN's requirements. For a solution of such NP-hard problem, we propose, in this paper, JSAR, an algorithm that combines simultaneously a network duty cycle scheduling policy for energy saving and so network lifetime maximization, a frequency channels and time slots allocation strategy for an efficient sharing of frequency channels and time slots during routing process, and an heuristic for a QoS support offering guaranteed services in compliance with the application requirements and taking into account the network configuration. In this current work, we only focus on the performance evaluation of JSAR. The network performance evaluation under JSAR is our future work.

The rest of the paper is organized as follows: In section 2, we present the system model. Section 3 describes the principle of PMCMTP [1] with the use of JSAR. In section 4, we detail the proposed cross-layer design. In section 5, we evaluate JSAR performance by analyzing and commenting some simulation results.

2 System model and notations

We consider an WMSN composed of one coordinator located at the center of the network and a set of distributed sensor nodes acting as routers and/or sources organized in fully meshed topology (See second tier of the network architecture proposed in [11]). We can model the network as a weighted directed graph $G(V, E, w)$. We admit that the network supports five data flow priority levels from P_1 to P_5 (P_1 for the highest priority level). The control traffic refers to the network synchronization and resource allocation information. All network nodes are concerned by the beacon frames. So, a simple manner to quickly share such information between all nodes is broadcasting it from the coordinator which is more powerful than others nodes, and has no energetic constraint compared to others nodes. Also, the amount of control traffic is negligible in comparison to the data traffic so there is no need to balance such traffic over the network. A single frequency channel will be enough to ensure the control traffic and the rest of available frequency channels can be used to maximize the number of parallel data communications. For those reasons, we propose one hop control communications using one frequency channel. To balance both load and energy consumption and to minimize power transmission, we propose multi-hop routing with the use of J frequency channels of K time slots per channel for data communication between network's nodes. Each node $u \in V$ has a unique identifier (id), and it is characterized by its initial battery capacity (BC_u), residual energy (Re_u) and data queue state ($DataQ_u$). A node presents three activity states respectively busy (transmitting or receiving data), forced

sleep (see section 3.A) and free (in sleep).

Notations:

V : set of vertices (nodes), E : set of edges (link)

N : number of nodes, L : number of links, w : link's weight, P_i : flow priority levels,

J : number of frequency channels, K : number of time slots per channel

dc_u : refers to a node duty cycle or activity state, $dc_u = 1$ for sleep or free state

$dc_u = 0$ for busy and in forced sleep states

Each link $l_{uv} \in E$ is associated with multiple parameters which can be classified into: additive metric (as delay, link cost, energy consumption, number of hops), multiplicative metric (as packet reception rate PRR) and concave metric (as available bandwidth [6]).

For non-additive metrics, we propose to prune all links that do not satisfy the constraints of concave metrics and for multiplicative metrics we transform it into additive metrics by applying a logarithm operation on it. For the rest of the paper, we consider only additive metrics for path evaluation. Each link $l_{uv} \in E$ is associated with a cost parameter $cost(u, v)$ and m additive nonnegative weights $w_i(u, v) \geq 0$, $i = 1 \dots, m$ which can be denoted by the vector $w(u, v) = \langle w_1(u, v), \dots, w_m(u, v) \rangle$ and each weight is corresponding to a constraint, and the upper bound of the i^{th} constraint is denoted by C_i (i.e vector $C = \langle C_1, \dots, C_m \rangle$ corresponds to the vector of the m constraints).

At a given time t , each link $l_{uv} \in E$ is characterized by a set of parameters as follows:

- Link's logarithmic value of packet reception rate prr_{uv} : We define prr_{uv} as the absolute value of the logarithmic value of PRR_{uv} (packet reception rate).

$$prr_{uv} = |\log_{10}(PRR_{uv})| \quad (1)$$

- Link's delay δ_{uv} : For each data-flow (F), the wireless interface delay would typically involve the data transfer delay (δ_{tr}^F), the resource allocation and the channel access delay (δ_{ch}^F). δ_{uv} is expressed in terms of time slots as follows:

$$\delta_{uv} = \delta_{tr}^F + \delta_{ch}^F \quad (2)$$

- Link's energy consumption E_{uv}^c : We define E_{uv}^c as the sum of the energy consumed by node u during radio transmission and data processing and queueing (E_u^c) and the energy consumed by node v during radio receiving (E_v^c).

$$E_{uv}^c = E_u^c + E_v^c \quad (3)$$

- Link's residual energy Re_{uv} : We define Re_{uv} as the minimum value of residual energy of nodes u and v .

$$Re_{uv} = \min(Re_u, Re_v) \quad (4)$$

- Link's energy availability Ae_{uv} : We define Ae_{uv} as a boolean value reflecting the energy availability for both nodes u and v .

$$Ae_{uv} = \begin{cases} 1 & \text{if } Re_u > E_u^c \text{ and } Re_v > E_v^c \\ 0 & \text{otherwise} \end{cases}$$

- Link’s available time slots TS_{uv} : We assume the use of K equal time slots (τ) with J frequency channels (i.e. $K \times J \tau$ in total). We consider TS_u^j (resp. TS_v^j) to represent the available time slots of the node u (resp. v) of the j^{th} channel and TS_{uv}^j to represent the link’s available time slots of the j^{th} channel. We model TS_u^j (resp. TS_v^j) as a binary vector of dimension K :

$$TS_u^j = \langle tsu_{j,k} \rangle_{1 \leq k \leq K} \quad tsu_{j,k} = \begin{cases} 1 & \text{if time slot } \tau_{j,k} \text{ is available} \\ 0 & \text{else} \end{cases} \quad (5)$$

$\tau_{j,k}$ represents the k^{th} time slot of the j^{th} channel.

$$TS_{uv}^j = \langle tsu_{j,k} * tsv_{j,k} \rangle_{1 \leq k \leq K} \quad (6)$$

At a given time t , we characterize a path $P_{s \rightarrow d}$, from a source s to a destination d , by a set of parameters as follows:

- Path’s packet reception rate $prr_{s \rightarrow d}$: We transform this multiplicative metric into a cumulative metric by applying a logarithm operation on it.

$$PRR_{s \rightarrow d} = \prod_{l_{uv} \in P_{s \rightarrow d}} PRR_{uv} \Rightarrow prr_{s \rightarrow d} = \sum_{l_{uv} \in P_{s \rightarrow d}} prr_{uv} \quad (7)$$

- Path’s delay $\delta_{s \rightarrow d}$: We define path’s delay as the sum of delay expired per each link in the path.

$$\delta_{s \rightarrow d} = \sum_{l_{uv} \in P_{s \rightarrow d}} \delta_{uv} \quad (8)$$

- Path’s energy consumption $E_{s \rightarrow d}^c$: We define path’s energy consumption as the sum of energy consumed per each link in the path.

$$E_{s \rightarrow d}^c = \sum_{l_{uv} \in P_{s \rightarrow d}} E_{uv}^c \quad (9)$$

- Path’s residual energy $Re_{s \rightarrow d}$: We define path’s residual energy as the minimum of residual energy of the set of all links in the path.

$$Re_{s \rightarrow d} = \min_{l_{uv} \in P_{s \rightarrow d}} Re_{uv} \quad (10)$$

- Path’s energy availability $Ae_{s \rightarrow d}$: We define path’s energy availability as the minimum of energy availability of the set of all links in the path.

$$Ae_{s \rightarrow d} = \min_{l_{uv} \in P_{s \rightarrow d}} Ae_{uv} \quad (11)$$

3 PMCMTP with the use of JSAR

In [1], we have proposed the PMCMTP for frequency channels and time slots allocation inside a personal area network. A key concept in PMCMTP is the elementary active cycle as shown in Figure 1, which is composed of two consecutive superframes, the first

for synchronization and collection of resource requests, and the second for the request scheduling algorithm (RSA), the reception of the second beacon and the data communications. In this paper, we propose the extension of the PMCMTP MAC protocol by including the JSAR instead of RSA to ensure the access to the medium, the duty cycle scheduling of network's nodes, the efficient sharing of available frequency channels and time slots and the search of the best routing paths in compliance with multiple application QoS requirements. Let us assume that the coordinator maintains a global vision of the network configuration. Whenever there is a change in the network, the coordinator updates related parameters (matrix and graphs...). The principle of PMCMTP using the JSAR is based on the following four phases:

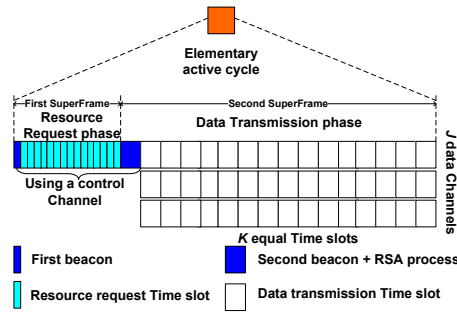


Fig. 1. An elementary active cycle

- **Forced Sleep test and synchronization:** Let us assume that the WMSN's coordinator maintains an updated energetic graph G_e of network's nodes. Before the start of each elementary active cycle, the coordinator executes a Force Sleep Test (FST) to exclude nodes with low levels of energy from participating in the routing process. It must include in the beacon payload the id of nodes that must go to forced sleep state. By listening to the first beacon, WMSN's nodes adjust their wake-up clocks and check the beacon payload to know if they must go to forced sleep state. Only nodes that pass the FST will be active during the current elementary active cycle and the others must go to forced sleep state to save their batteries.
- **Requesting resources:** This phase represents a set of equal short time slots, during which, the coordinator is listening to the requests of its members. Based on the number of source nodes, the coordinator assigns to each source node a specific time slot according to their id . Just following the reception of the first beacon, each source node waits for its own time slot to send its resource allocation request. The request packet is composed of five fields including the request identifier, the request's priority index, the number of required time slots per hop, the source and destination addresses.
- **Routing and resource allocation process:** According to the JSAR, after reception of all resource allocation requests, the coordinator schedules them according to their priorities. Once the list of requests are scheduled, the coordinator launches the routing process, an heuristic algorithm based on Dijkstra algorithm, to find the best path, then it ensures the effective allocation of time slots and frequency channels. For each request, it tries to find the adequate available time slots per available

channels to assign it for the entire selected path from the source to the destination taking into account QoS constraints and the resource availability. After processing all resource allocation requests, the coordinator computes the satisfactory note SN of the current active cycle, then it registers a trace of requests, which were not served to analyze it during the next cycle. Then, it inserts into the next beacon frame the necessary information of the served requests as the index of allocated channel, the index of the first allocated time slot, the number of allocated time slots and source and destination addresses.

- **Data transmission:** After listening to the second beacon, nodes can have a feedback of different requests. Each concerned sensor switches to the suitable channel at the suitable time slot and it begins sending or receiving data frames during the allocated duration.

4 WSN's cross-layer design

In this section, we propose a new WMSN's cross-layer framework, with joint consideration of the network's duty cycle scheduling policy, the resource allocation management strategy and the network layer QoS routing process to simultaneously ensure energy saving, optimal resource sharing and QoS support. The MAC layer has to support the duty cycle scheduling and the spectral and temporal resource allocation whereas the Network layer has to ensure the multi-constrained QoS routing process.

4.1 Duty cycle scheduling

The objective of the network's duty cycle scheduling is to maximize the node's sleep duration in order to prolong the network's lifetime while ensuring QoS requirements in a dynamic manner (i.e by balancing load and enhancing energy saving without affecting to the network performance). The dynamic Sleep/Wake scheduling problem in WMSN can be formulated as follows: *how should the activity of nodes be scheduled in a WMSN in order to enhance energy efficiency while ensuring network's QoS requirements?*

To determine which subset of nodes to be turned off for a time interval and the manner of sleep interval scheduling are both in joint correlation with routing process and application requirements. In order to minimize and balance energy consumption and to avoid the node's failure due to draining of battery, we suggest forcing certain routers to enter into sleep state. The nodes, whose residual energy is lower than the threshold value TH^n and their data queues ($DataQ_i$) are empty, are turned into forced sleep state during the n^{th} elementary active cycle (See Alg.1). TH^n is defined as follows:

$$TH^n = \frac{\sum_{i=1}^N Re_i^n}{N} * SN^{n-1}, \quad SN^{n-1} = \frac{\text{Number of satisfied routing requests}}{\text{Total number of routing requests}} \quad (12)$$

N and SN^{n-1} refer respectively to the number of WMSN's nodes and the satisfactory note of the $n - 1^{th}$ active cycle ($SN^{n-1} \in]0, 1[$). The satisfactory note SN^{n-1} is computed at the end of the $n - 1^{th}$ active cycle.

Let us assume that the duration of the n^{th} data communication phase is equal to K time slots with the use of J frequency channels. We model the duty cycle during data communication phase of the node i by the vector DC_i^n of dimension K .

$$DC_i^n = \langle dc_{i,k}^n \rangle_{1 \leq k \leq K} \quad dc_{i,k}^n = \begin{cases} 0 & \text{in forced sleep or busy} \\ 1 & \text{if free or in sleep} \end{cases} \quad (13)$$

The duty cycle matrix DC^n gives an idea about the activity states of the WMSN's nodes. This matrix can be updated during routing process (i.e. by putting the adequate time slots to 0, for each node participating in a path).

$$DC^n = \begin{pmatrix} DC_1^n \\ \vdots \\ DC_N^n \end{pmatrix}_{NK} = \begin{pmatrix} dc_{1,1}^n & \dots & dc_{1,K}^n \\ \vdots & \ddots & \vdots \\ dc_{N,1}^n & \dots & dc_{N,K}^n \end{pmatrix}_{NK} \quad (14)$$

Algorithm 1 : Forced Sleep Test, FST

for $i = 1$ to N {
 if ($Re_i^n < TH^n$ and $DataQ_i$ is empty) then
 $DC_i^n = \langle 0 \rangle$, update($G(V_a, E_a, w^n)$)}

DC_i^n : State of the i^{th} node for the n^{th} elementary active cycle

V_a, E_a : The active vertices and edges of the n^{th} active cycle

4.2 Resource Allocation

The objective of the resource allocation process is to optimally allocate available resource in terms of time slots per available frequency channels to ensure QoS support and to enhance both energy efficiency and channel utility. As Sleep/Wake scheduling problem, the resource allocation is also in joint correlation with routing process and application requirements. We model the resources of the the n^{th} data communication phase by the binary matrix RS_{JK}^n . Initially, $RS_{JK}^n = I_{JK}$ (i.e. initially there are in total $K \times J$ available time slots) which is updated during routing process.

$$RS_{JK}^n = \begin{pmatrix} rs_{1,1}^n & \dots & rs_{1,K}^n \\ \vdots & \ddots & \vdots \\ rs_{J,1}^n & \dots & rs_{J,K}^n \end{pmatrix}_{JK} \quad rs_{j,k}^n = \begin{cases} 1 & \text{if the } k^{th} \text{ slot of the } j^{th} \text{ channel is free,} \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

For a given link $l_{uv} \in E$, we compute the link's available time slots TS_{uv} , for the n^{th} active cycle, as given in Alg.2.

In order to maximize the resource utility and to optimally share such resource between WMSN's nodes, we suggest to implement a centralized resource allocation policy at the network coordinator. The resource allocation decision is taken by the coordinator in response to the set of collected resource allocation requests, according to the network configuration, the duty cycle scheduling and the routing decision. As shown in

Fig.2, first, the coordinator tries to find candidate paths according to the application requirements and network organization. During this process, it ensures the temporary resource allocation (allocation of the most earlier available time slots of available frequency channel per link) to compute path's parameters and cost, then it selects the best path. At this moment, the coordinator makes the effective resource allocation and it updates the resource matrix RS , the duty cycle matrix DC and the energetic graph G_e (i.e. by decreasing the residual energy, according to the energy consumption estimation, for each node participating in the path). We propose TRR (*Temporary Resource Reservation*) algorithm (Alg.3) to ensure the temporary resource reservation of the current path that can be added as extensible or candidate path. This algorithm consists to find the suitable available time slots per available channels to temporary assign it to the new edge l_{uv} . The main objective of this algorithm is to minimize, as possible, the end to end delay.

Algorithm 2 : Computation of the available time slots per link

for the n^{th} active cycle
 $\forall l_{uv} \in E$ do {
 for $j=1$ to J {
 Compute TS_u^j (resp. TS_v^j) of node u (resp. v)
 $TS_u^j = \langle dc_{u,k}^n * rs_{j,k}^n \rangle_{1 \leq k \leq K}$
 Compute $TS_{uv}^j = \langle tsu_{j,k} * tsv_{j,k} \rangle_{1 \leq k \leq K}$
 $TS_{uv} = \langle TS_{uv}^j \rangle_{1 \leq j \leq J}$

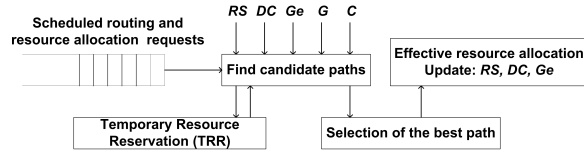


Fig. 2. Resource allocation algorithm for WMSN

Algorithm 3 : Temporary Resource Reservation: TRR

Given $\langle p, RS_{temp}, DC_{temp}, Ge_{temp} \rangle, l_{uv}$
 $\delta_{uvtemp} = \infty, ch=0, fts=0$
 Compute TS_{uv}
 if $(Ae_{uv} = 1)$ then
 for $j=1$ to J {
 Compute δ_{uv}
 if $\delta_{uv} < \delta_{uvtemp}$ then
 $\delta_{uvtemp} = \delta_{uv}, ch = j, fts = \text{First allocated } TS$
 compute $w(u, v)$, update $(RS_{temp}, DC_{temp}, Ge_{temp})$
 else $w(u, v) = \infty$

4.3 Multi-constrained QoS routing

The objective of QoS routing is to find a path from a source s to a destination d using the minimum amount of network resources such as energy consumption per route, residual battery power per node, available time slots, while satisfying various QoS constraints, such as delay, reliability, etc. When multiple routing metrics are considered, the problem becomes a multi-constrained path problem, which has been mathematically proven in [7] to be NP-complete. However, various heuristic and approximation algorithms, proposed in the literature, can approximate or solve similar problems in polynomial or pseudo-polynomial time [8, 9]. For WMSNs, we propose a centralized multi-constrained QoS reactive routing, which jointly interacts with the proposed duty cycle scheduling and resource allocation methods. The algorithm *JSAR*, given by Alg.4, describes the proposed routing approach.

Let $\mathbb{P}_{\mathbb{G}}$ and $\mathbb{P}_{s \rightarrow d}$ represent respectively the set of all paths and the set of all paths from a source node s to a destination node d in the network. $\mathbb{P}_{\mathbb{G}} = \bigcup_{\forall s, d \in V} \mathbb{P}_{s \rightarrow d}$.

The basic routing problems can be defined as follows:

Definition 1. *The MCP (Multi-Constrained Path) problem is to find paths $P_{s \rightarrow d}$ from s to d such that:*

$$w(P_{s \rightarrow d}) = \sum_{l_{uv} \in P_{s \rightarrow d}} \begin{pmatrix} w_1(u, v) \\ w_2(u, v) \\ \cdot \\ \cdot \\ w_m(u, v) \end{pmatrix} \leq \begin{pmatrix} C_1 \\ C_2 \\ \cdot \\ \cdot \\ C_m \end{pmatrix} \quad (16)$$

The goal is to find the set of feasible paths $\mathbb{P}_{s \rightarrow d}^f$ from s to d that satisfy multiple constraints simultaneously.

Definition 2. *The MCOP (Multi-Constrained Optimal Path) is one variant of MCP problem which is to find a path $P_{s \rightarrow d}^{opt}$ from s to d such that:*

$$Cost(P_{s \rightarrow d}^{opt}) = \min_{\forall P_{s \rightarrow d} \in \mathbb{P}_{s \rightarrow d}^f} Cost(P_{s \rightarrow d}) \quad (17)$$

The goal is to find the optimal feasible path from s to d in such a way that network resources are efficiently utilized.

Definition 3. *Let $P_{s \rightarrow d}$ and $P'_{s \rightarrow d}$ two different paths from s to d . Path $P_{s \rightarrow d}$ is dominated by path $P'_{s \rightarrow d}$ if and only if $w(P'_{s \rightarrow d}) \leq w(P_{s \rightarrow d})$ (i.e. $P_{s \rightarrow d} \prec P'_{s \rightarrow d}$) with at least one strict inequality.*

Lemma 1. *if $\exists P_{s \rightarrow d} \in \mathbb{P}_{s \rightarrow d}^f \implies \exists$ a non-dominated $P'_{s \rightarrow d} \in \mathbb{P}_{s \rightarrow d}^f$.*

According to Lemma (1), we can use the concept of path domination to reduce the computational complexity by considering only non-dominated feasible paths, given that the set of non-dominated feasible paths from s to d represents a sub set of $\mathbb{P}_{s \rightarrow d}^f$. We define $\mathbb{P}_{s \rightarrow d}^{fnd}$ as the set of all non-dominated feasible paths in $\mathbb{P}_{s \rightarrow d}^f$.

Lemma 2. Given $P_{s \rightarrow d} \in \mathbb{P}_{s \rightarrow d}^{fnd}$, let $P_{u \rightarrow v} \subset P_{s \rightarrow d} \implies P_{u \rightarrow v} \in \mathbb{P}_{u \rightarrow v}^{fnd}$ (*Optimality principle: Every optimal path is formed from optimal sub-paths*).

According to Lemma (2), a sub-path of a non-dominated feasible path is even a non-dominated feasible path. We consider only non-dominated feasible sub-paths to be extended to reach last destinations. Sub-optimal paths (i.e, paths that are dominated by or are equal to others) are ignored during path's extension process. Let two paths P and P' to vertex u and P is dominated by P' , if path P can be extended to a path that satisfies Eq. 16, then so also can P' . So, there is no need to retain P for path extension.

The determination of all non-dominated paths (i.e $\mathbb{P}_{s \rightarrow d}^{fnd}$) from s to d becomes a very hard task to accomplish when the number of such paths is too high. In this context, we propose to limit the search of non-dominated feasible paths for a set of at most X paths ($\mathbb{P}_{s \rightarrow d}^{Candidate}$). The main sub-problems that we must study are as follows:

- How to define the parameter X taking into account its impact on the routing decision and consequently on the network performance,
- How to evaluate a path,
- How to evaluate and select the set of the X most efficient non-dominated feasible paths if they exist.

Once the set $\mathbb{P}_{s \rightarrow d}^{Candidate}$ is obtained, one still faces the problem of selecting the best final solution.

Definition of the parameter X : To improve the efficiency of the path finding algorithm, X should be large, but it is to the detriment of a computation complexity. This parameter must be carefully determined in order to ensure a tradeoff between routing efficiency and both space and computation complexity. We propose two methods to compute X :

- Statically: The coordinator defines X as a fixed number computed as soon as the beginning of the network operation. We can use the definition used in [8].
- Dynamically: For each active cycle, the coordinator can dynamically compute X according to the network configuration, the supported load, and the satisfactory note of the previous active cycle.

Evaluation of a path: First, we propose a link's cost function. For resource allocation requests with the priority's level equals to P_1 or P_2 (i.e. hard real time constraint), we consider the delay as the metric of performance. Then, for resource allocation requests with the priority's level equals to P_3 or P_4 , we propose the energy consumption per link as the metric of performance given that energetic aspect becomes constraining. Finally, for the rest of requests we propose the residual energy per node as the metric of performance to avoid premature death of some nodes.

$$Cost_{uv} = \begin{cases} \delta_{uv} & \text{if } P = P_1, P_2 \\ E_{uv}^c & \text{if } P = P_3, P_4 \\ Re_{uv} & \text{if } P = P_5 \end{cases} \quad (18)$$

To ensure jointly QoS support, load balancing and energy saving, we propose the link's cost function (Eq.18) and the path's cost function (Eq.19) based on the resource allocation request's priority. For requests with priority P_1 and P_2 , the path cost represents respectively the end to end delay and the average delay. For requests with priority P_3

and P_4 , the path cost represents respectively the average energy consumption per path and the total energy consumption per path. Finally for last requests, in order to avoid network partition, the cost represents the residual energy per path.

Evaluation and selection of most efficient non dominated X paths: According to the relax operation (See Alg.4), each new generated path $((current\ path\ p) \cup l_{uv})$ must pass the feasibility test then the non-dominance test. In this case, the new founded path is stored, according to its cost, in the suitable position at the scheduled queues $PATH(v)$ (i.e $\mathbb{P}_{s \rightarrow v}^f$) and Q (non-dominated paths). The length of queues can not exceed X , otherwise, the last path must be discarded.

$$Cost_{s \rightarrow d} = \begin{cases} \sum_{l_{uv} \in P_{s \rightarrow d}}^{v=d} Cost_{uv} \text{ if } P = P_1 \\ \frac{\sum_{l_{uv} \in P_{s \rightarrow d}} Cost_{uv}}{N_{hop}} \text{ if } P = P_2 \\ \frac{\sum_{l_{uv} \in P_{s \rightarrow d}} Cost_{uv}}{N_{hop}} \text{ if } P = P_3 \\ \sum_{l_{uv} \in P_{s \rightarrow d}} Cost_{uv} \text{ if } P = P_4 \\ \min_{l_{uv} \in P_{s \rightarrow d}}(Cost_{uv}) \text{ if } P = P_5 \end{cases} \quad (19)$$

N_{hop} : the number of hops per a path $P_{s \rightarrow d}$ from s to d .

Selection of the best path: The best path can be selected among a set of candidate paths. The selection's criterion is the path's cost function. So, the minimum cost path will be selected as the best path. Consequently, the effective resource allocation takes place.

4.4 JSAR

As already explained, for each active cycle, the *JSAR* (See Alg. 4) performs firstly the Forced Sleep Test (line 2 in the main). After the reception of all resource allocation requests, it computes the parameter X (line 3 in the main). For each request it tries to find the set of candidate paths (line 4 in the main) then, from this set it selects the best path and ensures the effective resource allocation (line 5 in the main). Finally it computes the satisfactory note of the current cycle (line 6 in the main).

5 JSAR evaluation

We have implemented the proposed algorithm in MATLAB environment. The performance of the algorithm is evaluated by experiments on an ANSNET [10] network (Table.1). We consider a three-constrained path problem. Link weights are as follows:

- $w_1(u, v)$: represents the parameter prr_{uv} randomly selected from uniform $[0, 1]$,
- $w_2(u, v)$: represents the parameter δ_{uv} assumed be equal to 1 Time slot,
- $w_3(u, v)$: represents the parameter E_{uv}^c assumed be equal to 40 .

Constraints are randomly generated and five different ranges of the constraint vector C ($\langle C_1, C_2, C_3 \rangle$) are selected for each experiment (Table.2). The result is based on 100 randomly generated resource allocation requests for each range. The first experiment is carried out to test how X value affects the performance of JSAR. We compared JSAR with the exact algorithm where the length of path queue is infinite. According to Table.3, we note that the SN increases as the constraints get looser. Also, the SN increases with the increase of X . When X is equal to 5, JSAR is almost as good as the exact one. According to Fig.3, we observe that the execution time of JSAR increases with the increase of X , also it increases as the constraints get looser. Globally, the execution time of JSAR is acceptable for typical applications for its on-line implementation. The second experiment is carried out to respectively test how the number of available channels and the number of time slots affect the performance of JSAR. According to Table.4 and Table.5, in the majority of cases, the SN increases with the increase of respectively the number of available channels and the number of time slots. Respectively, the increase of the number of available channels and the number of time slots can significantly increase the performance of the algorithm especially when the constraints get looser. According to the network configuration and the application requirements (e.g. desired SN), those results can be used in practice to compute the necessary resources.

Table 1. Network configuration

Parameter	Default value
Topology	ANSNET model
Number of Vertices - Edges	32-54
Number of Time Slots per data communication phase	15
Number of Channels per data communication phase	3
Length of a path queue or X	5
Number of routing and resource allocation requests	100

Table 2. Range of C_1 , C_2 and C_3 for each experiment

Case	Range of $C = \langle C_1, C_2, C_3 \rangle$
1	$C_1 \sim \text{uniform}[1, 2], C_2 \sim \text{uniform}[1, 3], C_3 \sim \text{uniform}[40, 120]$
2	$C_1 \sim \text{uniform}[2, 3], C_2 \sim \text{uniform}[3, 6], C_3 \sim \text{uniform}[120, 240]$
3	$C_1 \sim \text{uniform}[3, 4], C_2 \sim \text{uniform}[6, 9], C_3 \sim \text{uniform}[240, 360]$
4	$C_1 \sim \text{uniform}[4, 5], C_2 \sim \text{uniform}[9, 12], C_3 \sim \text{uniform}[360, 480]$
5	$C_1 \sim \text{uniform}[5, 6], C_2 \sim \text{uniform}[12, 15], C_3 \sim \text{uniform}[480, 600]$

6 Conclusion

In this paper, we presented a new cross-layer design for WMSNs based on multi-channel MAC protocol. We have decomposed the optimization problem into three sub-problems: duty cycle scheduling to save energy and extend the network lifetime, frequency channels and time slots allocation to optimally allocate resource and enhance the network's performance and multi-constrained QoS routing to ensure QoS support

and save energy. We solved the three problems jointly and using *JSAR* that simultaneously satisfies multiple QoS requirements and ensures the optimal resource allocation. The evaluation results demonstrate the performance of *JSAR*. As future work, we will implement the proposed algorithm in a network simulator to evaluate the performance of the network in terms of end to end delay, throughput, lifetime and resource utility.

References

1. J. Ben Slimane, Y.Q .Song, A. Koubaa, "A Prioritized Multi-Channel Multi-Time slot MAC Protocol For Large-Scale Wireless Sensor Networks", COMNET'09, 2009, Tunisia.
2. J. Luo, L. Jiang, C. He, "Cross-Layer Optimization for Energy-Timeliness Tradeoff in TDMA Based Sensor Networks ", IEEE Global Telecommunications Conference, 2008, pp.1-5.
3. F. Bouabdallah, N. Bouabdallah, R. Boutaba, "Cross-Layer Design for Energy Conservation in Wireless Sensor Networks", IEEE International Conference on Communications, 2009, pp.1-6.
4. B. Yuebin, L. Shujuan, S. Mo, L. Yang and X. Cong, "An Energy Optimization Protocol Based on Cross-Layer for Wireless Sensor Networks", Journal of communications, 2008, pp.27-34.
5. L. Gang and K. Bhaskar, "Minimum latency joint scheduling and routing in wireless sensor networks", Journal Ad Hoc Networks, 2007, pp.832-843.
6. Dr.S Upadhyaya, G. Dhingra, "Exploring Issues for QoS Based Routing Algorithms", International Journal on Computer Science and Engineering, Vol 2, Iss 5, 2010, pp.1792-1795.
7. Z. Wang and J. Crowcroft, "QoS routing for supporting resource reservation", IEEE Journal on Selected Areas in Communications, 1996, vol. 14, pp.1228-1234.
8. X. Wendong et al, "An efficient heuristic algorithm for multi-constrained path problems", The 56th IEEE Vehicular Technology Conference, 2002, pp.1317-1321.
9. S. Wan, Y. Hao, Y. Yang, "Approach for Multiple Constraints Based Qos Routing Problem of Network", The 9th International Conference on Hybrid Intelligent Systems, 2009, pp.66-69.
10. D.E. Comer, Internetworking with TCP/IP, 3rd Edition, New York:Prentice Hall, 1995.
11. J. Ben Slimane, Y.Q .Song, A. Koubaa, M. Frikha, "A Three-Tiered Architecture for Large-Scale Wireless Hospital Sensor Networks", MobiHealthInf 2009, pp 20-31.

Table 3. SN vs X

Case	Exact algorithm	JSAR with X=			
		2	3	4	5
1	0.15	0.15	0.15	0.15	0.15
2	0.43	0.42	0.42	0.42	0.43
3	0.62	0.58	0.59	0.61	0.62
4	0.78	0.68	0.73	0.76	0.78
5	0.85	0.72	0.80	0.84	0.85

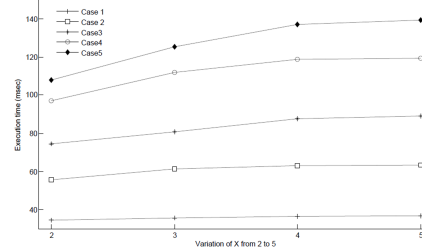


Fig. 3. Execution Time vs X

Table 4. SN vs N_{ch}

Case	JSAR with $N_{ch} =$			
	1	2	3	4
1	0.12	0.14	0.15	0.15
2	0.19	0.34	0.43	0.51
3	0.23	0.47	0.62	0.67
4	0.37	0.68	0.78	0.80
5	0.49	0.78	0.85	0.84

Table 5. SN vs N_{ts}

Case	JSAR with $N_{ts} =$		
	5	10	15
1	0.15	0.15	0.15
2	0.43	0.43	0.43
3	0.53	0.62	0.62
4	0.53	0.78	0.78
5	0.53	0.78	0.85

Algorithm 4 JSAR

Initialization($G(V_a, E_a, w), Rq, RS^n, DC^n, G_e^n$)
1 $s=Rq.s, d=Rq.d, RS_{temp} = RS^n, DC_{temp} = DC^n, G_{etemp} = G_e^n$
2 $\forall v \in V_a \setminus \{s\} \text{ } PATH(v)=\emptyset$
3 $PATH(s)=\{\langle \vec{0}, RS_{temp}, DC_{temp}, G_{etemp} \rangle\}$
4 $Q = \mathbb{P}_{s \rightarrow d}^{Candidate} = \{\langle \vec{0}, RS_{temp}, DC_{temp}, G_{etemp} \rangle\}$

X efficient PATHs($X, \langle q, RS_{temp}, DC_{temp}, G_{etemp} \rangle, Queue$)
1 $\forall p \in Queue$
2 **if** $Cost(q) < Cost(p)$ **then** {
3 *Insert* $\langle q, RS_{temp}, DC_{temp}, G_{etemp} \rangle$ *from* p 's position in *Queue*
4 **if** $|Queue| = X + 1$ **then**
5 *Delete last path in Queue*

Relax($p, v, RS_{temp}, DC_{temp}, G_{etemp}, C$)
1 **TRR**($\langle p, RS_{temp}, DC_{temp}, G_{etemp} \rangle, l_{uv}$)
2 **if** $W(p \cup l_{uv}) > C$ **then**
3 **return**
4 *dominated*=*False*
5 $\forall q \in PATH(v)$
6 *Compute* $W(p \cup l_{uv})$
7 **if** $p \cup l_{uv} \succ q$ **then**
8 *dominated*=*True*
9 **if** $p \cup l_{uv} \prec q$ **then**
10 *Remove* $\langle q, RS_{temp}, DC_{temp}, G_{etemp} \rangle$ *from* $PATH(v)$
11 *Remove* $\langle q, RS_{temp}, DC_{temp}, G_{etemp} \rangle$ *from* Q
12 **if** (\neq *dominated*) **then**
13 $\langle q, RS_{temp}, DC_{temp}, G_{etemp} \rangle = \langle p \cup l_{uv}, RS_{temp}, DC_{temp}, G_{etemp} \rangle$
14 **X efficient PATHs**($X, \langle q, RS_{temp}, DC_{temp}, G_{etemp} \rangle, PATH(v)$)
15 **X efficient PATHs**($X, \langle q, RS_{temp}, DC_{temp}, G_{etemp} \rangle, Q$)

Routing($G(V_a, E_a), Rq, RS^n, DC^n, G_e^n$)
1 **Initialization**($G(V_a, E_a), Rq, RS^n, DC^n, G_e^n$)
2 **While**($Q! = \emptyset$ and $|\mathbb{P}_{s \rightarrow d}^{Candidate}| < X$) {
3 $\langle p, RS_{temp}, DC_{temp}, G_{etemp} \rangle = Dequeue(Q)$
4 $u = last_{element}(p)$
5 **if** ($u = d$) **then**
6 *Inqueue* $\langle p, RS_{temp}, DC_{temp}, G_{etemp} \rangle$ *in* $\mathbb{P}_{s \rightarrow d}^{Candidate}$
7 **else** $\forall l_{uv} \in E_a$
8 **Relax**($p, v, RS_{temp}, DC_{temp}, G_{etemp}, C$) }
9 **return** $\mathbb{P}_{s \rightarrow d}^{Candidate}$

1 **The Best PATH**($\mathbb{P}_{s \rightarrow d}^{Candidate}, RS^n, DC^n, G_e^n$)
2 $\langle p_{best}, RS_{temp}, DC_{temp}, G_{etemp} \rangle = \min_{\forall p \in \mathbb{P}_{s \rightarrow d}^{Candidate}}(Cost(p))$
3 $RS^n = RS_{temp}, DC^n = DC_{temp}, G_e^n = G_{etemp}$

Main
1 **For** the n^{th} active cycle **do** {
2 **Forced Sleep Test**
3 **X Computation**($G(V, E), DC^n, RQ^n, SN^{n-1}$)
4 **for** $i = 1$ to $|RQ^n|$ **do** {
5 $\mathbb{P}_{s \rightarrow d}^{Candidate} = Routing(G(V_a, E_a), RQ[i], RS^n, DC^n, G_e^n)$
6 **The Best PATH**($\mathbb{P}_{s \rightarrow d}^{Candidate}, RS^n, DC^n, G_e^n$) }
7 *Compute* SN^n }
