



HAL
open science

On the interest of opportunistic anycast scheduling for wireless low power lossy networks

Thong Huynh, Fabrice Theoleyre, Won-Joo Hwang

► To cite this version:

Thong Huynh, Fabrice Theoleyre, Won-Joo Hwang. On the interest of opportunistic anycast scheduling for wireless low power lossy networks. *Computer Communications*, 2017, 104, pp.55-66. 10.1016/j.comcom.2016.06.001 . hal-02565973

HAL Id: hal-02565973

<https://hal.science/hal-02565973v1>

Submitted on 6 May 2020

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.

©2017

This work is licensed under a Creative Commons “Attribution-NonCommercial-NoDerivatives 4.0 International” license.



On the Interest of Opportunistic Anycast Scheduling for Wireless Low Power Lossy Networks

Thong Huynh^a, Fabrice Theoleyre^b, Won-Joo Hwang^{a,*}

^a*Department of Information and Communication System, HSV-TRC, Inje University, 197, Gimhae, Gyeongnam 621-749, Korea*

^b*ICube Laboratory, CNRS / University of Strasbourg, Pole API, Boulevard Sebastien Brant, 67412 Illkirch Cedex, France*

Abstract

Low-Power and Lossy Networks (LLNs) aim at integrating smart objects into the Internet of Things. IEEE 802.15.4-TSCH is currently a promising standard for the link layer: it schedules the transmissions and implements slow channel hopping to improve the reliability while the routing layer focuses on constructing distributed routes for a small collection of destinations (i.e. convergecast). We propose here an efficient scheduling policy to exploit an opportunistic feature of the MAC layer: a single transmission is received by a collection of next hops which decide opportunistically which one will forward the packet. We consider here the problem of the optimal scheduling policy for reliability and energy efficiency when considering such opportunistic forwarding at the MAC and routing layers. The simulation results demonstrate the effectiveness of the proposed policy: by effectively selecting the set of parents (i.e. next hops) and carefully considering the channel quality, the energy consumption per packet is reduced. Besides, we also improve the reliability: the network can also use unreliable radio links, where only one of the next hops receives the packet to forward. This scheduling policy may typically be implemented in the PCE of IEEE 802.15.4e-TSCH.

Keywords: opportunistic routing, multi-parent, scheduling, multichannel MAC, IEEE 802.15.4e TSCH.

1. Introduction

Recent advances in technology made possible the creation of smart objects that can be interconnected to create the new Internet of Things. The IEEE 802.15.4 working group proposed in 2012 an amendment to enable energy efficient networking for the industrial market [1]. Since transmissions are often predictive, the TSCH (Timeslotted channel Hopping) mode proposes to assign timeslots coupled with a slow channel hopping strategy to improve the reliability. To enable an energy efficient TDMA, TSCH implements an implicit synchronization: any data or control packet may be used by the receiver or the transmitter to compute the clock drifts. This strategy is particularly efficient for periodic and/or predictive transmissions.

In multihop networks, a centralized Path Computation Engine (PCE) may compute the path to use for each flow. Alternatively, a routing protocol such as RPL may construct distributively the routes [2]. For a convergecast traffic, most of the approaches construct a Directed Acyclic Graph rooted at the border routers

*Corresponding author. Tel.: +82-55-320-3847.

Email addresses: huythongtc@gmail.com (Thong Huynh), theoleyre@unistra.fr (Fabrice Theoleyre), ichwang@inje.ac.kr (Won-Joo Hwang)

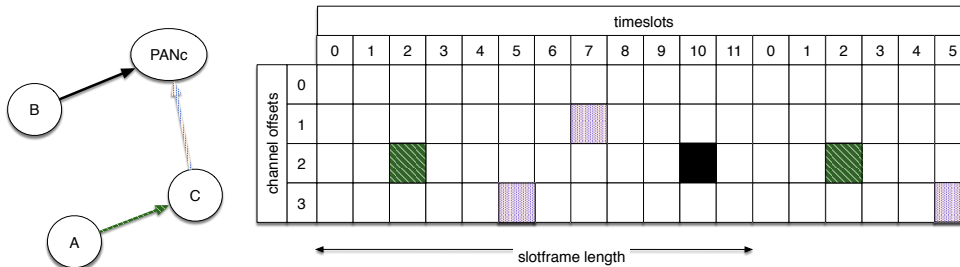


Figure 1: Structure of the IEEE 802.15.4-TSCH superframe

(i.e. gateways to the Internet). Then, each node has a set of parents, which constitute the next hops toward the border routers. However, RPL selects currently only one parent, which forwards all the traffic – the other ones serve just as a backup purpose. We propose here to modify the MAC forwarding strategy to enable opportunistic forwarding, exploiting all the parents to improve the reliability. Indeed, losing packets may be prejudicial for many applications since packets may not be redundant (e.g. water metering, smart parking, etc.)

Scheduling the transmissions after having selecting the routes to use has recently attracted much attention. In particular, the 6TiSCH working group [3] aims at defining the mechanisms to enable RPL to work on top of IEEE 802.15.4 TSCH. 6TiSCH defines a bootstrapping procedure with a minimal configuration so that a central controller (the Path Computation Engine – PCE) is then able to compute an efficient schedule. However, 6TiSCH currently focuses on unicast transmissions, with the concept of *tracks*. We aim at going further in this direction, with an opportunistic version of IEEE 802.15.4-TSCH, exploiting several parents in parallel to increase the reliability while decreasing the energy consumption. Typically, our solution may be implemented in 6TiSCH working with a centralized controller (PCE).

Proposing an efficient schedule also requires to determine the optimal transmission power. Indeed, using the maximum power may improve the reliability but may also increase the energy consumption [4]. Besides, the channel quality varies with time due to fading. Thus, we can achieve significant energy conservation by scheduling transmission during the time that the channel is in good quality [5].

In this paper, we consider the problem of scheduling in multihop convergecast networks to improve both the end-to-end reliability and the global energy consumption. In particular, we use the multiple parent opportunistic scheduling at the MAC layer to cope with individually faulty links. The contributions of this paper are twofold:

1. We provide a dynamic and stochastic formulation of the problem by using dynamic programming. This policy can maximize a combination of reliability and energy consumption. We propose an approximate algorithm, which considers the problem in close form function rather than discrete value, to reduce the computational complexity;
2. We thoroughly evaluate the performance of this algorithm by simulations. Our scheme results in a lower energy consumption and a better end-to-end reliability compared to a non optimized, single-parent scheduling strategy (Section 6).

2. Related Work

2.1. IEEE 802.15.4e-TSCH

IEEE 802.15.4-2006 [6] was introduced at first in Personal Area Networks (PANs). While it enabled through Zigbee the adoption of IEEE 802.15.4 for the Internet of Things (IoT), it suffers from limits in multihop topologies [7], and leads to many collisions [8].

The working group proposed in 2012 the IEEE 802.15.4-TSCH amendment [1] to set-up an industrial wireless network TSCH adopts a FTDMA strategy such as Wireless HART [9] to mitigate interference and multipath fading. Besides, TSCH is deterministic to provide predictable performance.

In IEEE 802.15.4-TSCH, a schedule is established so that each node knows at the beginning of each timeslot if it has to stay awake to receive or transmit a frame. In the IEEE 802.15.4-TSCH jargon, a *cell* represents a transmission opportunity, denoted by a timeslot and a channel offset. We denote in this paper by cell the pair $\langle \text{timeslot}, \text{channelOffset} \rangle$.

The superframe structure is depicted in Fig. 1. A slotframe contains a certain number of timeslots and repeats over time. In a timeslot fits only one data packet, which may be acknowledged by the receiver. A slot may be either dedicated to one transmitter or shared among a group of interfering nodes. In Fig. 1, the radio link $C \rightarrow PAN_c$ reserved two cells (timeslots 5 and 7) for its transmissions.

TSCH may use a centralized scheduling, with a Path Computation Engine (PCE). This PCE is the central entity in charge of computing the paths used by each flow, and the channel and timeslots used by each radio link. In that case, the 6TiSCH architecture is an extension of the detnet work Deterministic Networking Architecture [10], adopting a SDN approach.

The schedule may also be decentralized, decided one-by-hop with the 6top protocol to negotiate a group of cells [11]. Because each pair of nodes selects autonomously the cell to use, collisions may arise. Some mechanisms are required to monitor and adapt the local schedule [10].

Interference may arise in the following cases:

internal interference occurs if the same cell is allocated to a pair of interfering transmitters. Since no medium access is implemented, the collision will occur for sure in *every* cell.

We assume here the interfering topology is a priori known, and that our scheduling algorithm forbid to allocate the same cell to two different interfering transmitters. We neglect consequently this internal interference.

external interference may come from other wireless technologies (Bluetooth, Wifi, etc.) since we use the ISM band [12, 13]. However, channel hopping is particularly efficient to alleviate this kind of external interference, reducing the number of retransmissions [14].

We consider here that external interference impact negatively the Packet Error Rate (PER): more cells have to be allocated when external interference is present. Besides, we focus on long-term performance, i.e. we consider external interference is sufficiently stable to be estimated via the PER, and thus considered in the scheduling process.

2.2. Routing

RPL [2] was designed for low power lossy networks (LLN), mainly for convergecast traffic, i.e. all the packets are collected by a border router.

The standard constructs a Destination Oriented Directed Acyclic Graph (DODAG): each node maintains a collection of parents, and the routing structure does not form any cycle. In the forwarding plan, a node chooses to forward all its packets to its preferred parent, other parents constitute backup solutions if the preferred one fails.

To construct a DODAG, each node computes a *rank*, denoting its *virtual distance* from the border router. Then, a node selects as parent any node with a lower rank (Fig. 2).

Lampin *et al.* [15] proposed an opportunistic forwarding version of RPL to exploit also long radio links with a poor reliability. The parent with the higher priority acknowledges first the packet, They use an opportunistic MAC layer, where a single transmission is sufficient to forward a packet to all the parents. Opportunistically, the parent with the higher priority and which received the packet acknowledges the reception and forwards the packet. **We want to propose here a scheduling policy to exploit this opportunistic feature in a 6TiSCH stack.**

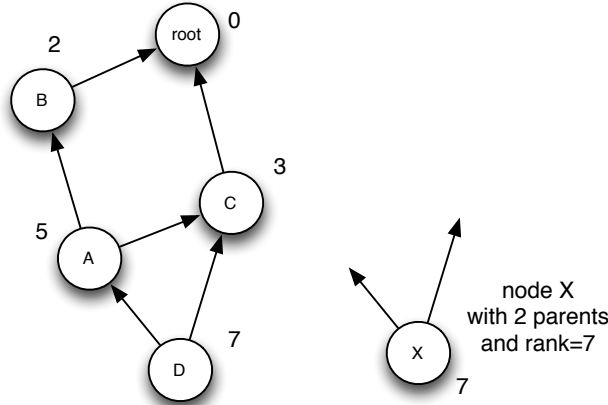


Figure 2: DODAG structure of RPL

2.3. FTDA Scheduling Algorithm

To assign the resource for each packet is currently a very challenging objective: we have to define which *cell* should be used for each radio link.

Ghosh *et al.* [16] proposed to minimize the schedule length in a multichannel TDMA environment. Tsitsiklis *et al.* [17] studied the tradeoff between a centralized and a distributed scheduling. By adopting a queue theory based approach, they demonstrated a centralized approach is more efficient. Thus, these approaches are well suited for industrial networks with strict requirements on reliability and delay.

Yang *et al.* [18] constructed an optimal schedule for time-sensitive flows: new cells are inserted in the schedule if the end-to-end reliability is insufficient until the deadline constraint is not fulfilled. TASA proposes to construct a centralized scheduling for a multihop TSCH network [19]. Yigit *et al.* [20] studied the impact of routing on the scheduling: using unreliable links increases the number of timeslots required to achieve a minimum reliability. Dobsław *et al.* [21] proposed to reserve additional timeslots for retransmissions.

DeTAS proposed a decentralized version of TASA [22]: the children of the border routers collect the information and compute the schedule of their subtree (called micro-schedules). Finally, the micro-schedules are re-arranged into a globally acceptable schedule. Distributed scheduling is particularly recommended when the traffic or the topology are dynamic, where the network may temporarily perform poorly before the network re-converges to a legal state.

In this paper, we propose to explore rather how opportunistic forwarding at the MAC layer may improve the reliability while limiting the energy consumption. This multi-parent feature may be incorporated in these scheduling algorithms.

3. Opportunistic Multi-Parent Forwarding with TSCH and 6TiSCH

We propose to modify TSCH to implement opportunistic forwarding: a node transmits its data frames in *anycast* toward *any* of its parents.

3.1. Opportunistic Forwarding with IEEE 802.15.4-TSCH

For an opportunistic TSCH, several parents must wake-up at the beginning of a dedicated cell. A child transmits its data frame in anycast, and the parent with the largest priority and which received correctly the packet must send an acknowledgement. This method is similar to those adopted by Lampin *et al.* [15].

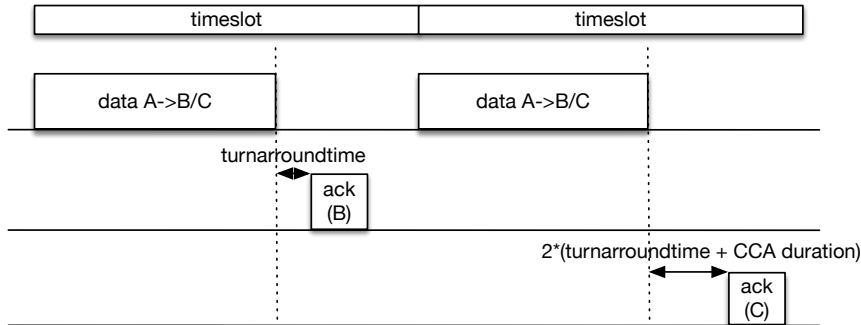


Figure 3: Opportunistic Multiparent Transmission

After the data packet, the remaining part of the timeslot is dedicated to the opportunistic acknowledgements. The parents are ranked by the child: the source gives a preference order among its parents. If a parent does not hear any acknowledgement for a sufficiently long time, it decides to forward the packet.

More precisely, the k^{th} parent triggers k successive CCA, interspaced by a `turnarroundtime` (maximum time to switch from RX to TX mode). Since the parents are strictly ordered, we can see that a parent stops deferring as soon as it receives an ack.

To reduce the overhead, we upper bound the number of possible *active* parents in a given timeslot. Practically, 4 parents are sufficient to offer a certain diversity [23]. Given the CCA duration ($128\mu\text{s}$) and the `turnarroundtime` ($210\mu\text{s}$), approximately 1ms is dedicated for the acknowledgments at the end of the timeslot.

The parents must hear each other to listen to the acks of others. To construct the schedule, a node should report the link quality toward its neighbors. Thus, the PCE will select in a given timeslot a set of parents with a very small PER. Besides, a parent will consider the packet is acknowledged if the CCA fails, i.e. it does not need to decode the `ack` packet, it just needs to sense a radio activity above the CCA threshold. Practically, missing the `ack` is very unfrequent. Finally, missing an acknowledgement means only duplicates are generated, consuming a little more resource.

Figure 3 illustrates the timeslot structure for the radio links (A,B) and (A,C) for the topology depicted in Fig. 2. We assume that the parent *B* is preferred over the parent *C*. In the first timeslot, the parent *B* answers first, and *C* will stop listening to the medium. In the second timeslot, *B* did not receive correctly the packet, and *C* will acknowledge after having deferred sufficiently. A transmission fails only if **all** the parents are unable to decode the packet.

Such feature is particular useful since unreliable links are the rule in realistic environments [24] and the radio link quality may even evolve along the time [25].

3.2. 6TiSCH

We aim in this paper at proposing a policy to schedule efficiently dedicated opportunistic cells: during one time a child transmits a frame to several parents. We aim at implementing our scheduling policy inside the Path Computation Element (PCE).

Thus, we expect to re-use entirely the 6TiSCH protocols. The PCE:

- collects globally the statistics from each participant (RSSI, LinkQuality, Queue status, etc.) with the 6TiSCH Resource Management features [26];
- uses consequently the network topology (link quality) to compute the schedule in a centralized manner;
- pushes each local schedule to each node, using end-to-end CoAP messages [26].

Table 1: Notation

Symbol	Meaning
P_i	the parents of the node i
$x_{i,j}(t)$	the node i may transmit a packet to j during the timeslot t
$z(i)(t)$	the amount of packets left in the buffer of node i at time t
D_i	the number of packet in the buffer node i need to transmit during the cycle
T	number of timeslot in each cycles
$s_{i,j}(t)$	channel state for the transmissions from the node i to its parent j
Γ_i	activation vector of node i (i.e. set of parents scheduled during this timelot)
Γ	feasible set of activation vectors (i.e. interference free)
$I_{i,j}(\cdot)$	the packet is correctly delivered from i to its parent j
$e_{i,j}(\cdot)$	energy consumed for a transmission from i to its parent j
$V^{(\pi)}(\cdot)$	value function of the ADP (i.e. total reward from the beginning)
$\bar{V}^{(k)}(\cdot)$	approximate value function of the ADP at iteration k

4. System Model and Problem formulation

We consider the convergecast traffic problem in wireless sensor networks with one border router (root) and N sensor nodes. The sensor nodes report periodical measures (e.g. pollution, noise, gas or water consumption) to the root node. All sensor nodes are battery-powered with finite reserve. In our model, the DODAG structure is constructed by using a routing protocol such as RPL so that each node can forward its data to the root by using its parents. Our objective consists in determining a set of activate nodes allowed to transmit data, and its associated parents such that the transmission energy of overall network is minimized under condition of maximizing the total amount of receiving data.

4.1. System Model

Let $i \in N$ represent the index of the sensor node and P_i represent the set of its parents. Two nodes can communicate with each other if they are mutually in their radio range (i.e. the reception power is sufficient to decode the frame). We consider a slotted time frame structure that is used in IEEE 802.15.4-TSCH. The time is divided into equal cycles (slotframes). Furthermore, each cycle is composed of T timeslots and a timeslot is the basic unit time for transmission (fig. 1). At the beginning of each cycle, the scheduling policy will decide the set of nodes allowed to transmit, based on the energy budget of each sensor node within a cycle.

4.1.1. Channel Model

We model the channel condition by using a Finite-State Markov Channel (FSMC) model. This FSMC models accurately Rayleigh fading channel [27]. Each node may use some dedicated beacon frames to estimate the signal-to-noise ratio (SNR) on the radio link. We divide the range of possible SNR values into equal intervals where each interval represents a state in the Markov chain. We denote the set of the states by $s^{(w)} \in S = \{s^{(0)}, s^{(1)}, \dots\}$. Let $s_i(t) = \{s_{i,j}\}_{i \in N, j \in P_i}$ be the vector representing the channel state of node i , where $s_{i,j} \in S$ is the channel states of radio link between node i and its parent j . The channel states of sensor nodes among these states $s_i(t)$ follow the Markov processes. The successive states $s_{i,j}(t+1)$ of link (i, j) at $t+1$ are governed by the transition probability $p_{i,j}^{(w,k)} = Pr[s_{i,j}(t+1) = s^{(k)} | s_{i,j}(t) = s^{(w)}]$.

These transition probabilities are estimated through the probability density function (PDF) of the SNR. Each Markov state represents the interval of the SNR. From this function, we can estimate the states transition probability in Markov model as in [28]. This probability is equal to the probability that the SNR lies inside the given interval.

4.1.2. Interference model

Let $x_i(t) = \{x_{i,j}\}_{i \in N, j \in P_i}$ be a vector representing the activation state of node i where $x_{i,j} = 1$ if node i transmits data to its parents $j \in P_i$, otherwise $x_{i,j} = 0$. In case we use the Opportunistic Multi-Parent approach, each node may forward data to its parents simultaneously by using the timeslot structure as depicted in Fig. 3.

To determine which radio links may be scheduled simultaneously, we use the *contention graph* model [29]. Each vertex in the contention graph represents a radio link and a pair of radio links are neighbors in the contention graph (i.e. an edge exists) if they interfere with each other. We use here the interference range model [29] in order to construct the contention graph. If the distance between two nodes is shorter than the interference range, they cannot be scheduled at the same time without collision, and an edge exists in the contention graph. However, any other interference model is also relevant, it just has to be mapped in the contention graph.

Let Γ_i be the feasible activation vector of node i regarding its parents P_i and $\Gamma = \{\Gamma_i\}_{i \in N}$ is the feasible set in which all nodes in Γ can be active at the same time. Let G_C denote the contention graph. It is clear that a feasible set corresponds to an *independent set*, a set of radio links for which no edge between two arbitrary vertices exists in the contention graph. Finding the independent set is widely known as a NP-hard problem. However, there are several techniques to reduce the complexity [30]. Moreover, for a specific network topology, we need to compute the *independent set* only once at the beginning. We here adopt the heuristic algorithm in [29], to determine a subset of *independent sets*. It has been shown that the transmission rates under this heuristic algorithms are very close to those computed by using all possible transmissions.

The success/failure delivery of the packet from node i to its parent j is represented by an indicator transmission random variable $I_{i,j}(x_{i,j}(t), s_{i,j}(t))$ that depend on the channel states and the scheduling policy. Specifically, this indicator function is defined as follows:

$$I_{i,j}(x_{i,j}(t), s_{i,j}(t)) = \begin{cases} 1, & \text{if a packet is successfully} \\ & \text{received by } j \\ 0 & \text{otherwise} \end{cases}$$

Note that $I_{i,j}(x_{i,j}(t), s_{i,j}(t))$ is a random variable, and its conditional expectation given $(x_{i,j}(t), s_{i,j}(t))$ is equal to the success probability under a given channel quality. Thus, under specific channel states $s_{i,j}(t)$, activation vector, $I_{i,j}(x_{i,j}(t), s_{i,j}(t))$ can be a deterministic function 0/1 [31, 32].

Illustrative Example: Let us consider in Fig. 4 the two-states Markov channel, also known as the Gilbert-Elliot channel. In this channel model, each state corresponds to a specific channel quality which is either very good or totally bad. If the measured SNR is below the threshold value, the channel is labeled as "Bad" (B), otherwise, the channel is labeled as "Good" (G). The channel state is $S = \{G, B\}$. The transition probability from state i to state j is determined by $p^{i,j}$. Otherwise, each state associates with the success transmission probabilities $p_i, (i = G, B)$. The indicator transmission on radio link (i, j) , $I_{i,j}$, evolves over slots according to a random process with probability $p_i, (i = G; B)$. In our example, if the channel state is B , the transmission will fail ($p_B = 0$) and if the channel state is G , the probability of having successful transmission will be p_G .

4.1.3. Energy model

We make the common assumption that communication radios are the main source of energy consumption. This energy includes i) energy for transmitting packets, ii) energy for receiving packets. However, the total energy consumption for receiving a packet is fixed and does not depend on the channel states and the scheduling policy. Consequently, we neglect here this part.

Similarly to [33], we focus on optimizing the transmission energy. Let $e_{i,j}(x_{i,j}(t), s_{i,j}(t))$ be the transmitting energy of the user i at a specific channel state $s_{i,j}(t) = s^{(w)} \in S$, when it transmits a packet to its parent j . This energy can be calculated using the well-known Shannon capacity formula:

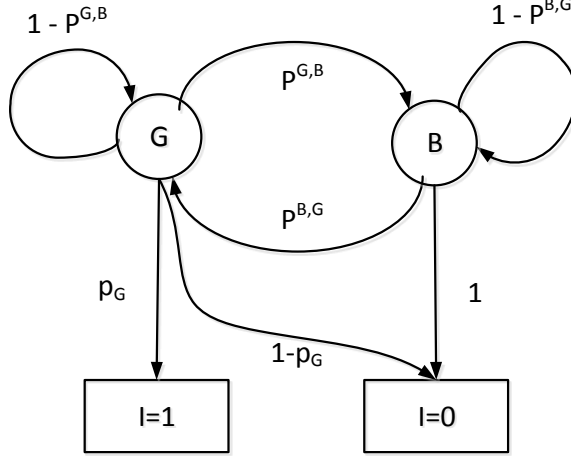


Figure 4: Illustration of a 2 states Markovian Channel model

$$e_{i,j}(x_{i,j}(t), s_{i,j}(t)) = x_{i,j}(t) \frac{N_0 W (2^{r/W} - 1)}{|h(t)^{(s_{i,j}(t))}|^2} \quad (1)$$

where N_0 is the additive Gaussian noise, r is the transmission rate, and W is the bandwidth. $h(t)^{(s_{i,j}(t))}$ is the channel gain between nodes i and j under the channel condition $s_{i,j}$. Since $e_{i,j}(x_{i,j}(t), s_{i,j}(t)) = 0$ if $x_{i,j}(t) = 0$, this allocation power is also a function of the activation vector. Let $e_i(t) = \max_{j \in P_j} e_{i,j}(t)$ be the transmission energy of node i at time t . It will ensure that all packets are transmitted with a sufficient power to reach all its parents. For notational convenience, in the rest of the paper, we use $e_{i,j}(t), I_{i,j}(t)$ instead of $e_{i,j}(s_{i,j}(t)), I_{i,j}(x_i(t), s_i(t))$; the dependence on $x_i(t), s_i(t)$ is implicit.

4.2. Problem formulation

To this end, each node i is associated with a utility:

$$U_i(t) = \sum_{j \in P_i} I_{i,j}(t) - \delta e_i(t) \quad (2)$$

that describes the balance between the throughput (i.e. number of received packets) and the allocated energy. The parameter δ controls the trade off between two objectives: the throughput maximization and the energy consumption minimization for each node.

We consider long-term network performance over one cycle length, which consists in T timeslots. Network performance is characterized by the network-wide utility of all sensor nodes over T timeslots. Each sensor node i has D_i packets which must be sent during the cycle. We would like to maximize the total utility

$$\max \sum_{t=0}^T \sum_{i \in N} U_i(t) \quad (3)$$

subject to constraints on the energy consumption, and the transmitted packet:

$$\sum_{t=0}^T \sum_{j \in P_i} e_{i,j}(t) \leq E_{m,i} \quad (4)$$

$$\sum_{t=1}^T x_i(t) = D_i \quad (5)$$

$$x_{i,j}(t) \in \{0, 1\}$$

The equation (4) limits the energy budget of each sensor node within a cycle. The constraint (5) ensures that each sensor node i will transmit D_i packets during the cycle. This problem belongs to the category of nonlinear integer programming, which is known to be NP-hard. Otherwise, it is challenging to find the optimal solution since the channel states of each user are uncertain, and the complexity of the problem which spreads over T timeslots and N nodes is increasingly high.

5. Dynamic Programming Framework

5.1. Original Dynamic Programming

This problem can be considered as a sequential decision problem with T stages. Let $E(t) = \{E_i(t)\}_{i \in N}$ be the energy remaining of all sensor nodes at time t . We define the variable $z(t) = \{z_i\}_{i \in N}$ that indicates the amount of packets left in the buffer at time t . The sensor node needs to transmit these packets by the end of the cycle.

We define a dynamic system in which states are the pair $S(t) = (E(t), z(t))$. The control action is the activation vector $x(t)$. At the beginning of each cycle, these states are measured and the scheduling policy chooses the action vector $x(t) \in \Gamma(t)$ to forward the system to the next stage. The transition function from the state at time t to the state at time $t + 1$ is defined by a function of $x(t)$ as follows:

$$\begin{aligned} E(t+1) &= E(t) - e(t)x(t) \\ z(t+1) &= z(t) - x(t) \end{aligned} \quad (6)$$

We also define the penalty cost $p = \{p_i\}_{i \in N}$ that are charged at the end of horizon T if a node i does not transmit completely D_i packets during the cycle. This cost is higher than the utility cost to ensure the scheduling policy *must* choose the control vector $x(t)$ to satisfy the constraint (5). This penalty cost function is computed at the end of the cycle as follows:

$$C(z(T)) = pz(T) \quad (7)$$

where $z(T)$ is the amount of packet left in the buffer at the end of the cycle.

We now calculate the one-period reward function $R(t) = \{R_i\}_{i \in N}$ under states $S(t)$ and activation vector $x(t)$ at each time t as follows:

$$R_i(t) = \begin{cases} U_i(t) & \text{if } t < T \\ U_i(T) - C(z(T)) & \text{if } t = T \end{cases} \quad (8)$$

As shown in equation (2), the utility depends on the current channel state. However, the current channel states at time t are random variables with distributions which may be computed from the one-step transition probability provided that the channel state at $t = 0$ is given [34, 27]. Let $p_{i,j}^{(w)}(t)$ be the probability that the radio link (i, j) undergoes the channel state $s^{(j)}$ at time t ; thus, $\sum_{s^{(j)} \in S} p_{i,j}^{(w)}(t) = 1$. The total expected utility at t is:

$$U(S(t), x(t)) = \sum_{i \in N} \sum_{s^{(j)} \in S} p_i^{(w)}(t) U_i^{(w)}(S(t), x(t)),$$

where $U_i^{(w)}(S(t), x(t))$ is the utility of node i with channel states $s^{(w)}$. This utility is computed as follows

$$U_i^{(w)}(S(t), x(t)) = \sum_{j \in P_i} I_{i,j}^{(w)}(t) - \delta e_i^{(w)}(t)$$

where $I_{i,j}^{(w)}(t)$, $e_i^{(w)}(t)$ are an indicator of the transmission random variable, and the energy consumed by the sensor i under channel states w .

The problem now becomes static: the expected reward function at each stage is a function only of the control decision $x(t)$ and the state $S(t)$. A policy $\pi = \{x(t)\}_{t=0,1,\dots,T-1}$ is a set of consecutive control actions generated from the states $S(0), S(1), \dots, S(T)$. The dynamic programming problem is formulated with the dynamic evolution of states described in (6) as

$$\max_{\pi} V^{(\pi)}(S(0)) = \left\{ R(S(T), x(T)) + \sum_{t=0}^{T-1} R(S(t), x(t)) \right\}, \quad (9)$$

where $(S(0))$ and $(S(T))$ are the initial and terminal states and the value function $V^{(\pi)}(S(t))$ is the total reward starting from t to T under the policy π . This function is equivalent to the negative of the cost-to-go function in control theory. Problem (9) is a formal representation of the association control problem in T stages. The problem consists in finding a control policy π so that the value function $V^{(\pi)}(S(t))$ (i.e. the total reward of the network) is maximized.

Let $V^*(S(t)) = \max_{\pi} V^{(\pi)}(S(t))$ be the optimal value function at stage t . We use '*' to denote the optimal setting, i.e. states and control actions. The relationship between $V(S^*(t))$ and $V^*(S(t+1))$ is given by

$$V^*(S(t)) = \max_{S(t), x(t)} \{R(S(t), x(t)) + V^*(S(t+1))\} \quad (10)$$

$V^*(S(t))$, the optimal control action $x^*(t)$ and state $S(t)$ may be computed recursively backward in time starting from $t = T - 1$ and stopping at $t = 0$. At the final step, an optimal scheduling policy π^* , which is defined by a set of actions $x^*(t)$, is generated.

Computation Complexity: It is important to note that the energy states are not discrete. We may divide the energy into quantified sample values. However, since we work with discrete quantities, solving the dynamic programming optimally becomes computationally intractable when the number of states and the decision variables exponentially increase.

The size of our states $S(t) = \{E(t), z(t)\}$ is $|E_m|^N 2^L$ where $|E_m|$ is the cardinality of the sample energy set. Thus, by using the backward induction algorithms to solve the dynamic programming, it requires $T |E_m|^N 2^N |\Gamma|$ to find the optimal activation vector [35]. This is the classical curse of dimensionality which is cited as the weakness of dynamic programming.

Consequently, we propose now to approximate the dynamic programming method, to obtain a complexity linear with the number of nodes.

5.2. Approximate dynamic policy (ADP): Value function approximation

Clearly, the time complexity increases with the number of nodes. The optimal policy suffers from the curse of dimensionality. Furthermore, calculating the backward cost is generally intractable, To alleviate this problem, we may use an approximate dynamic programming scheme.

In this approach, we propose to replace the value function $V(E(t), z(t))$ with a linear approximation. However, this approximation implies that this algorithm runs iteratively. We use the notation $[parameter]^k$

Algorithm 1: Approximate Dynamic Programming (ADP)

Step 0: Initialization.

Initialize $V^0(t) = 0$ for all t .

Set $k=1$.

Initialize $E^1(0) = \{E_{i,m}\}$, $z^1(0) = 1$.

Step 1: Do for $t = 0, 1, \dots, T$.

Step 1a: Solve.

$$x^k(t) = \arg \max_{x(t) \in \Gamma} \{R(S^k(t), x(t)) + \bar{V}^{k-1}(S^k(t+1))\}.$$

$$\tilde{V}^k(S^k(t)) \leftarrow \{R(S^k(t), x^k(t)) + \bar{V}^{k-1}(S^k(t+1))\}.$$

Step 1b: Update the states.

$E^k(t+1) \leftarrow E^k(t) - e^k(t)x^k(t)$.

$z^k(t+1) \leftarrow z^k(t) - x^k(t)$.

Step 1c: Update the value function.

$$\bar{V}^k(S^k(t)) = (1 - \alpha_k) \bar{V}^{k-1}(S^k(t)) + \alpha_k \tilde{V}^k(S^k(t)).$$

Step 2: Update iteration.

$k \leftarrow k + 1$.

if $k < K$ **then**

Go to step 1.

end if

Step 3:

Return the value function $\{V^K(t)\}_{t=1, \dots, T}$.

Return the activation vector $\{x^K(t)\}_{t=1, \dots, T}$.

to represent the specific value of [*parameter*] at iteration k . Let $\bar{V}(E(t), z(t))$ be an approximation of the value function. In the following, we demonstrate how to iteratively calculate the $\bar{V}(E(t), z(t))$.

During the iteration k , at time t , given the current states $(S^k(t))$, the decision $x^k(t)$ is computed by using the value function approximation $\bar{V}_{k-1}(S^k(t))$. This value function is computed from the previous iteration $k = 1, \dots, k - 1$ so it is indexed by $k - 1$. The scheduling variable $x^k(t)$ is given by solving

$$\tilde{V}^k(S(t)) = \max_{S^k(t), x^k(t)} \{R(S^k(t), x^k(t)) + \bar{V}^{k-1}(S^k(t+1))\}. \quad (11)$$

After getting the decision $x^k(t)$, the states $(E^k(t+1), z^k(t+1))$ is updated as in (6). Then the approximation value function at iteration k is calculated based on the moving average

$$\bar{V}^k(S^k(t)) = (1 - \alpha_k) \bar{V}^{k-1}(S^k(t)) + \alpha_k \tilde{V}^k(S^k(t)) \quad (12)$$

where $\{\alpha_k\}$ is the sequence that satisfies the following properties: $0 < \alpha_k < 1$; $\sum_{k=0}^{\infty} \alpha_k = \infty$; and $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$ to ensure the convergence [36]. The equation (12) is described as the exponential smoothing approximation. It can help to robust the approximation [37]. After reaching the end of cycle, we increase k and start over again.

Thus, beginning with the initial value function, we step forward in time. We solve the dynamic programming (10) approximately by using the value function approximations of the future states. More specifically, for each iteration k , we go through a horizon time, obtain the state trajectory and update the approximated value function for next iteration $k + 1$. We repeat this procedure over several iterations and update the value function $\bar{V}(E(t), z(t))$. The pseudo code for this approach is illustrated in Algorithm 1.

There are two interesting features of the ADP: First, we compute the value at each iteration by going forward in time instead of going backward as in the original dynamic programming approach. Thus, we can eliminate the step that required the looping over all the states. The ADP implementation uses consequently more efficiently the memory caches. Secondly, the approximate value function has a linear form. Indeed, our algorithm only adds the linear element to the original approximation. Thus, at each iteration, the approximate value function includes the original approximation and the linear element.

Lemma 1: The approximation function \bar{V}^k at iteration k is a linear function.

Proof. Let us prove this lemma by induction. We assume this lemma is true for the iteration $k - 1$: \bar{V}^{k-1} has linear form. We show it also holds k . From (12), we can write

$$\begin{aligned}\bar{V}^k(S^k(t)) &= (1 - \alpha_k) \bar{V}^{k-1}(S^k(t)) + \alpha_k \tilde{V}^k(S^k(t)) \\ &= (1 - \alpha_k) \bar{V}^{k-1}(S^k(t)) + \alpha_k R(S^k(t)) \\ &\quad + \alpha_k \bar{V}^{k-1}(S^k(t+1))\end{aligned}$$

Since the function $\bar{V}^{k-1}(S^k(t))$ and $R(S^k(t))$ have the linear form, the $\bar{V}^k(S^k(t))$ is a linear function. \square

5.3. Further Discussion

Computation Complexity: Since the approximated function \bar{V}^k has a linear form, the problem (11) corresponds to linear integer programming. In ADP algorithms, we need to solve the problem in (11), $KT(2N+1)$ times. This is linear with the number of nodes in the network, as opposed with the original dynamic programming. However, our approximate algorithm also converges to the optimal value when K is sufficiently large [38].

Message passing: At the beginning of the cycle, the sensor node measures the channel quality and determines the Markov states from itself to the parent. Then the sensor nodes send the message about the Markov states and the number of packet D need to be sent during the cycle to the PCE. The message can be of $\lceil \log_2(C+D) \rceil$ bits where C is the number of Markov states. The PCE computes the schedule as in the Algorithms 1. Then, it pushes the scheduling decision to each sensor node by sending T bits message to indicate the node defer or transmit data over T timeslots of the cycle. Since we limit the maximum number of active parent to 4, total $4N \lceil \log_2(C+D) \rceil + TN$ bits are required to compute the schedule at each cycle.

A node may send the channel state toward its parents in a CoAP message, using the 6TiSCH Resource Management features. In addition, the PCE can also broadcast the schedule using constrained application protocol (CoAP) with end-to-end CoAP message [26].

Since we consider the channel states are known at the beginning of the slotframe, our scheduling policy obtains an upper bound of what we may obtain without this knowledge. The schedule may provide a lower throughput if the channel states are not known by the PCE.

Our results show that opportunistic anycast scheduling improves the throughput in these conditions. In a future work, we aim at investigating how such anycast policy may be implemented practically while minimizing the overhead. Prediction Techniques may help to achieve such feature [39].

6. Performance Evaluation

In this section, we evaluate the proposed approximate dynamic policy (ADP) in comparison with a few existing policies under different performance criteria. We measure the following metrics:

Percent from optimal is the distance between the utility under ADP and under the original dynamic policy. This metric estimates the convergence of the ADP policy to the optimal value;

Throughput (packet/timeslots) is the average number of packets received successfully by the sink;

Energy consumption per packet (mW/packet) is calculated by taking a ratio of the total energy consumed by all the nodes to the total number of packets that are successfully received by the sink;

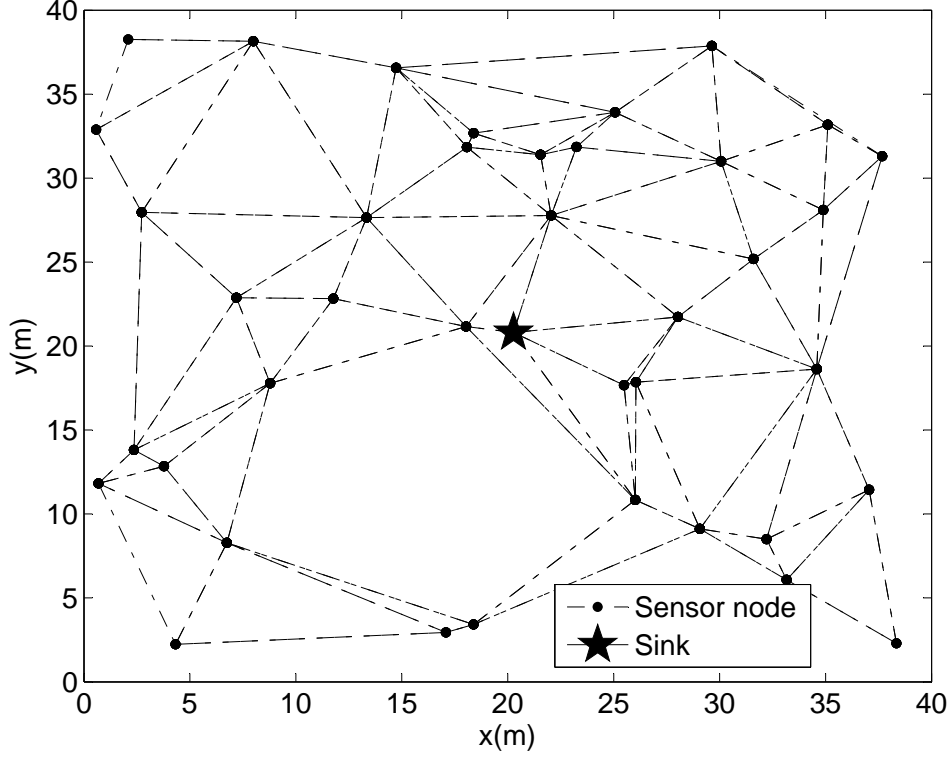


Figure 5: Random network topology used in evaluation

Packet error rate is the ratio between the number of lost packets and the number of transmissions. A packet may be lost either because of a queue overflow or a bad quality channel error.

We model the networks environment and conduct the simulation by using MATLAB simulator. We implemented the following scheduling policies for comparison:

1. **Conventional RPL policy (CRPL)**: This policy is a variant of our proposition. A node uses only one single parent to send all its traffic to the border router. This represents the usual behavior of RPL, forwarding the packets only to the preferred parent. The policy is:

$$V(S^*(t)) = \max \{R(S(t), x(t)) + V(S^*(t+1))\}$$

subject to

$$\sum_{j \in P_i} x_{i,j} \leq 1$$

This constraint guarantees that each node uses exactly one parent.

2. **Multi-Parent Greedy policy (MPaG)** is a sub-optimal policy, which selects the set of users with the maximum possible utility to transmit at a given time. It ignores the impact of the current optimal solution to the future cost function. The MPaG also uses the multi-parent scheme. At the timeslot t , we choose the user and its set of parents according to the following objective:

$$\arg \max \left\{ \sum_{i \in N} U_i(t) \right\}.$$

Table 2: Network parameter

Parameters	Values
# Sensor node and # sink	40 and 1
Simulation area	$40 \times 40m^2$
Transmission range	15m
Interference range	30m
Transmission rate (r)	250 kbps
Timeslot duration	10 ms
Data transmission duration	4.256 ms
Mini-timeslot duration	0.672 ms
# Max retransmission	3
# Max parents	4
Bandwidth (B)	20kHz
Noise (N_0)	-90dB
Energy budget for each cycle ($E_{m,i}$)	500mW
Tradeoff parameter (δ)	10 to 90

3. **Approximate dynamic programming policy (ADP)** is our proposition. Contrary to CRPL, the ADP considers all the parents. By using Algorithm 1, we solve the original dynamic programming as follows:

$$V(S^*(t)) = \max \{R(S(t), x(t)) + V(S^*(t+1))\}.$$

6.1. Simulation Environment

6.1.1. Network Scenario

We consider random network topologies (Fig. 5) where all the nodes send packets to the root node. We place randomly 40 nodes in a square area: the typical network diameter is around 7 hops. We use the interference range model [40]. The dot line in Fig. 5 shows the communication link of sensor nodes. We use expected transmission (ETX), a common routing metric in low power lossy networks [2]. A node has at most 4 active parents.

6.1.2. Channel model

We use the channel model introduced in Fig.4 to capture the channel condition. Different transition probabilities, $p^{i,j}$, reflect the channel quality. For example, if $p^{G,B}$ is high and $p^{B,G}$ is low, the channel quality is bad.

In addition, we model the external interference as the ON/OFF process and the transitions of the states follow the Bernoulli random variable with parameter p_{ext} . If the state is ON, the external interference causes the attenuation β to the signal strength of the link. The SNR is the product of the exponential random variable with parameter β and Bernoulli random variable with parameter p_{ext} . We set attenuation factor is 0.7. in the simulation.

6.1.3. Operation of the simulation

The simulation runs in a slotted time manner. At each simulation timeslot of (100 ms), the simulation executes the following steps. At the first step, the SNR is updated according to the channel model. Second, we update the energy and the success probability under the given channel condition. Finally, we run the scheduling policy using one of the three algorithms.

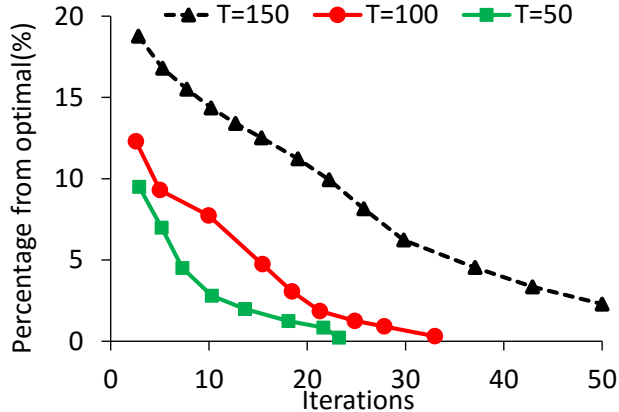


Figure 6: Convergence of the ADP policy

6.1.4. Simulation Parameters

The simulation parameters are listed in the Table 1, using IEEE 802.15.4e-TSCH [1], [41]. The transmission rate is 250 kps for the 2.4GHz band. The duration timeslot is 10ms. The maximum-length packet is 127 bytes which takes 4.256ms to transmit. Four mini-timeslots are used for ack in the end of timeslot. The duration of each mini-timeslot is 0.672ms which include : 192 μ s for turnaroundtime, 0.128 ms for ACC, and 0.352 ms for acknowledgment transmission.

6.2. Convergence

In the first scenario, we verify the convergence of the ADP policy by investigating the utility over 50 iterations. Fig. 6 shows the distance between the utility under ADP and the original dynamic policy which obtains the optimal utility. The convergence rate largely depends on the number of timeslots in a cycle: the larger value of T , the larger number of iteration to obtain the optimal solution.

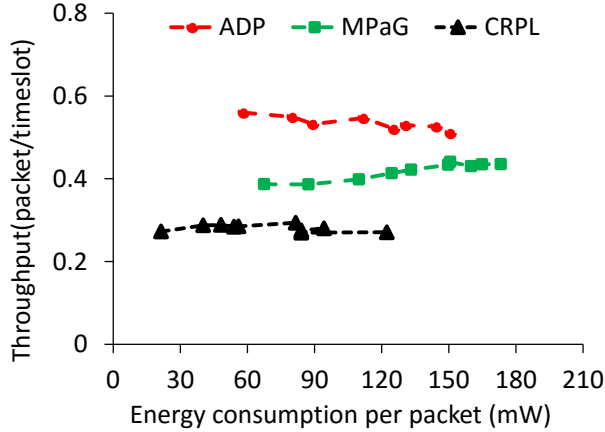
In general, the IEEE 802.15.4-TSCH proposes to use by default 100 timeslots in a cycle [1]. We are within 5 percent after 15 iterations and 1 percent after 30 iterations when the number of timeslots is 100. Thus, our ADP policy obtains a solution close to the optimal one even with a low number of iterations.

In the following scenarios, we set the number of iteration $K = 15$ and $T = 100$, where the percentage form the optimal solution is 5 %. It will ensure the convergence of the ADP policy.

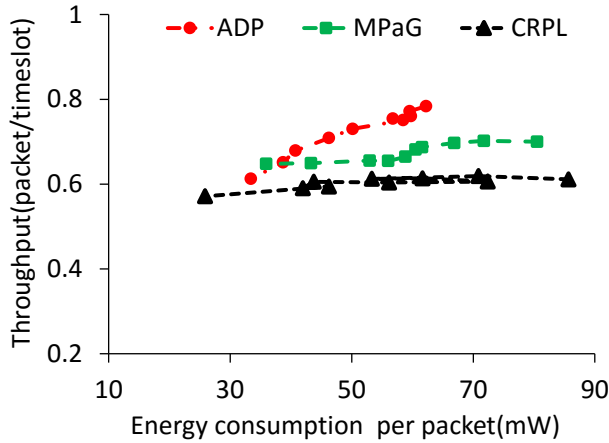
6.3. Tradeoff between throughput and energy consumption

We now investigate the impact of the δ control parameter (eq. 2). In the utility function, the throughput is about ten times different in value than energy consumption. Thus, we evaluate the power consumption and the throughput by varying the control parameter δ from 10 to 90. In Fig. 7, we show the Pareto curves, which reveal the dependency of the throughput and the energy consumption. Fig. 7(a) depicts the Pareto curves when the channel states have bad quality, $p^{G,B} = 0.8$. CRPL saves energy, but mainly because of reliability: many packets are not delivered to the sink, and do not *consume* energy. Indeed, exploiting one unique parent increases the probability the packet is not received correctly, even after a few retransmissions. We observe that ADP has the highest throughput and it has a lower energy consumption compared with MPaG. We guess this significantly higher throughput justifies the slightly larger energy consumption.

Similar results are shown in Fig. 7(b) where the channel qualities between each radio link are higher at $p^{G,B} = 0.2$. MPaG provides a better throughput than CRPL, since it has wider degrees of freedom for selecting multiple parents for every timeslot. Since ADP makes scheduling decision based on the expected rewards, disturbance of stochastic channel is thus reduced. Hence, its performance is better than MPaG. In the remaining scenarios, we set the control parameter $\delta = 50$ which is the middle point of these Pareto curves.



(a) Throughput and energy consumption under bad channel condition, $p^{G,B} = 0.8$.



(b) Throughput and energy consumption under good channel condition, $p^{G,B} = 0.2$

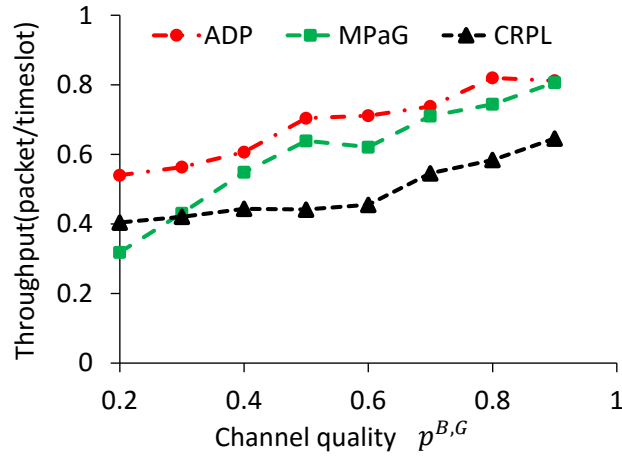
Figure 7: Throughput and energy consumption under different channel quality.

6.4. Impact of the channel quality

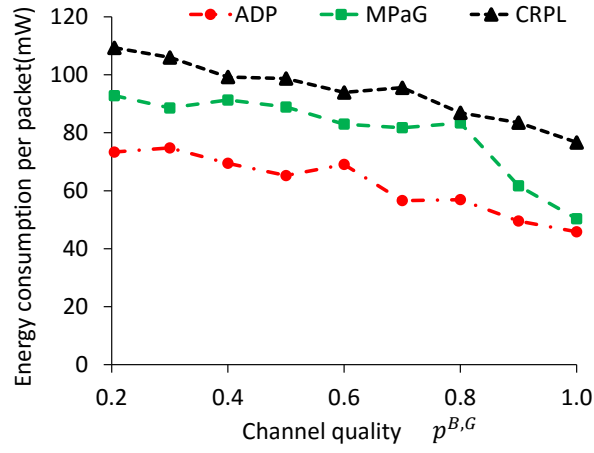
We investigate the performance of the policies under different channel qualities since they impact significantly the performance of the scheduling. Figure 8(a) and Figure 8(b) confirm the strong dependency with the channel quality. Obviously, the throughput of ADP, CRPL and MPaG increases with a better channel quality: less packets are lost, meaning also less retransmissions. However, It provides the highest throughput compared with CRPL and MPaG.

On other side, ADP is the most energy efficient strategy (fig. 8(b)): the quantity of energy spent by the network for each packet received by the sink is minimal. Indeed, using multiple parents helps to reduce the energy consumption by increasing the reliability. By effectively selecting several parents and carefully considering the channel quality, we can avoid the packet losses by distributing the traffic among several parents which have a good channel quality. In other words, CRPL and MPaG waste energy for packets which are not correctly received by the next hop because of unreliable links.

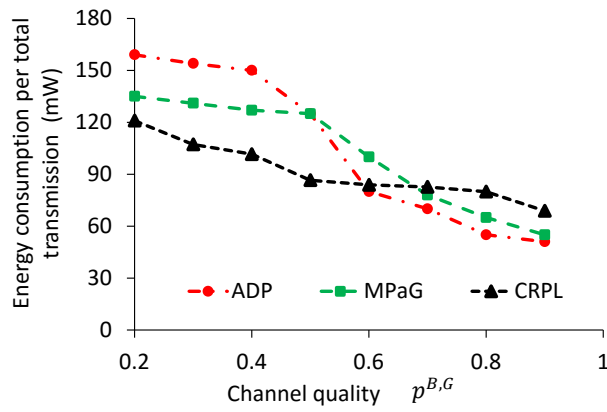
In addition, Figure 8(c) shows the energy per total transmissions. This energy is calculated by taking a ratio of the total energy consumed by all the nodes to the total number of transmissions that include both successful transmission and error transmission. ADP consumes the highest energy to transmit one packet



(a) Throughput

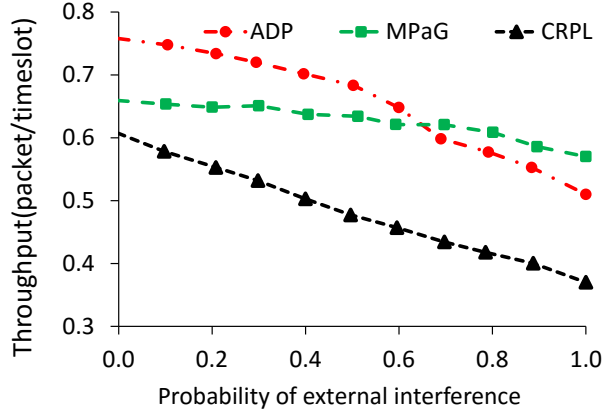


(b) Energy consumption per successful packet

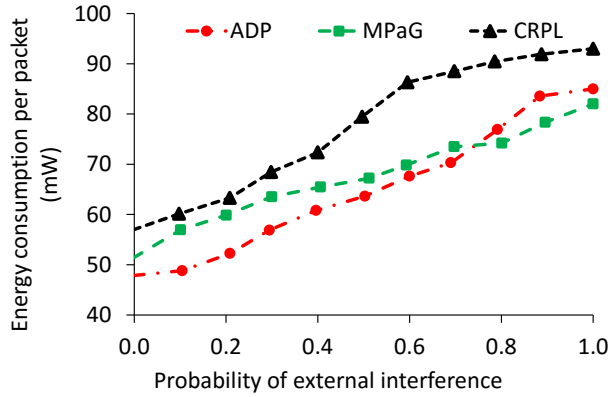


(c) Total energy consumption per transmitted packet (including dropped packets)

Figure 8: Throughput and energy consumption under variant channel quality, $p^{B,G}$.



(a) Throughput



(b) Energy consumption

Figure 9: Throughput and energy consumption under variant of timeslots in cycle, T .

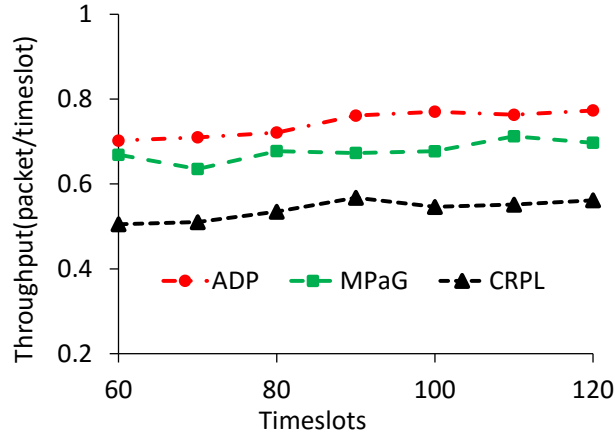
especially when the channel quality is bad. The main reason is that the child i needs to transmit with a sufficient power to reach all its parents $j \in P_i$: $e_i(t) = \max_{j \in P_i} e_{i,j}(t)$.

6.5. Impact of the external interference

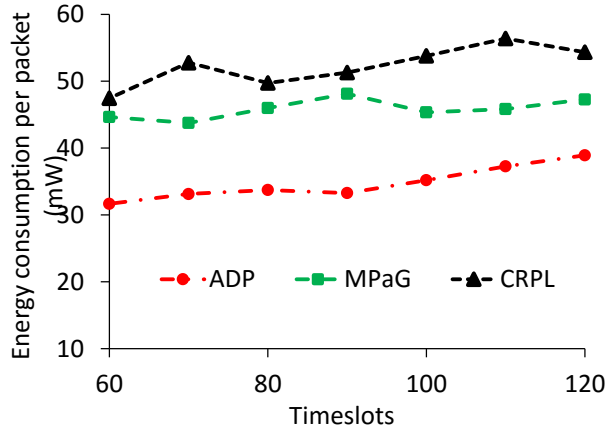
We investigate the performance of policies under the different probability that the external interference occurs. The results in Figure 9 have two interesting features. First, when external interference occur more frequently ($p_{ext} \geq 0.8$), the ADP cannot predict the behavior of the channel quality properly. Therefore, the performance of the ADP slightly decreases in comparison with the MPaG which use instantaneous channel quality to make decision. Second, the gap between the CRPL and the ADP, MPaG increase when more external interference occurs. By using multi-parent to forward data, both ADP and MPaG policies can combat with the external interference better than the CRPL which use single parent to forward data does.

6.6. Impact of the duty cycle ratio

We evaluate the impact of the number of timeslots in a cycle (Fig. 10). We vary the timeslot in a cycle from 50 to 150. The throughput slightly increases with a larger number of timeslots. Indeed, the throughput is proportional to the number of transmission opportunities, and the *pressure* reduces when the cycle (slotframe) is larger. In other words, we have the same quantity of traffic to transmit in a single slotframe while increasing the number of possible timeslots.



(a) Throughput



(b) Energy consumption

Figure 10: Throughput and energy consumption under variant of timeslots in cycle, T .

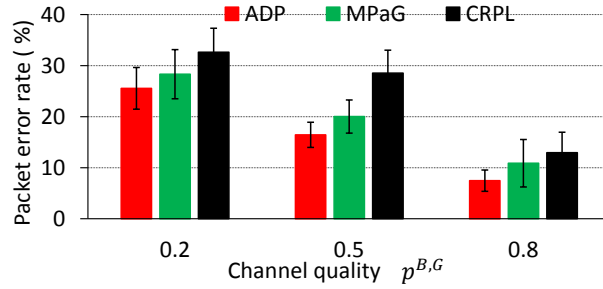
However, incrementing the number of timeslots does not significantly affect the energy consumption. This means that these policies, including ADP, are scalable with the number of timeslots in a cycle.

6.7. Reliability

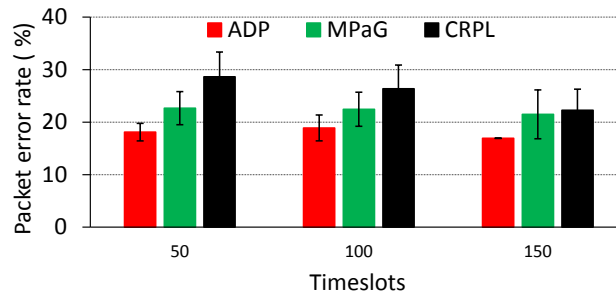
Finally, we provide the packet loss ratio for each policy under different channel qualities. Obviously, less packets are lost with reliable links (i.e. larger channel quality).

We can observe in Fig. 11 that ADP outperforms MPaG and CRPL concerning the packet loss rate. As we mention before, using multiple parents in ADP reduces the packet losses. Thus, ADP obtains a low energy consumption and an high throughput.

In addition, Fig. 11(b) illustrates the packet error rate when varying the duty cycle ratio. The error rate is only marginally impacted by the duty cycle ratio. We can remark that ADP performs significantly better than MPaG and CRPL when we have an high *pressure*, i.e. less timeslots to schedule all the traffic. Thus, our scheduling solution is efficient to schedule all the transmissions while limiting the energy consumption and presenting an high spectral efficiency.



(a) Packet error rate under different channel quality



(b) Packet error rate under different cycles

Figure 11: Packet error rate.

7. Conclusion

Channel Hopping MAC are considered one of the most promising solution for industrial networks. IEEE 802.15.4-TSCH proposes in particular to assign time-frequency blocks for each radio link, coupled with channel hopping to improve the reliability. We propose here a new scheduling policy to exploit an opportunistic forwarding scheme: during each timeslot, the next hop is dynamically chosen among the next hops which received correctly the packet. We propose consequently to improve the reliability since the transmission is successful if *any* of the parents correctly received the packet. A node may exploit in the same timeslot both unreliable and reliable links to improve the energy efficiency.

We proposes an Approximate Dynamic Programming (ADP) policy to schedule during each timeslot non-interfering links while optimizing the energy and the spectral efficiency. We formulate a policy using a Markovian model for the radio channel quality. Simulations results highlight the interest of such feature to improve both the reliability and the energy consumption.

While we demonstrated the interest of opportunistic forwarding, we have now to investigate the most accurate scenarios where this technique has to be adopted. In particular, a transmission requires several receivers to stay awake: because a node can listen to one channel at a time, we create a more constrained coloring problem. We aim at investigating the particular scenarios (i.e. for which radio links) this opportunistic strategy is the most recommended?

Besides, we assumed here the channel states are known at the beginning of the slotframe. How could we propose a schedule which relaxes this assumption? Could prediction algorithms help to solve this problem?

We also plan to investigate experimentally the performance of this strategy, implemented in a Path Computation Engine (PCE). We aim at providing also a traffic adaptive approach: when a new flow is accepted, the schedule is modified and the network reconfigured.

Acknowledgments

This work was partially supported by the PHC STAR 2012 program (Korean and French governments) under contract DCS and by the French National Research Agency (ANR) project IRIS under contract ANR-11-INFR-016

References

- [1] IEEE 802.15 WPAN task group 4b (tg4e), Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer (April 2012).
- [2] T. Winter, et al., RPL: IPv6 routing protocol for low-power and lossy networks, rfc 6550, IETF (2012).
- [3] 6tisch, Ipv6 over the tsch mode of ieee 802.15.4e wg, charter-ietf-6tisch-01-00 (2013).
- [4] A. Willig, Recent and emerging topics in wireless industrial communications: A selection, *IEEE Transactions on Industrial Informatics* 4 (2) (2008) 102–124. doi:10.1109/TII.2008.923194.
- [5] M. Zafer, E. Modiano, Optimal rate control for delay-constrained data transmission over a wireless channel, *Information Theory, IEEE Transactions on* 54 (9) (2008) 4020–4039. doi:10.1109/TIT.2008.928249.
- [6] IEEE 802.15 WPAN task group 4b (tg4b), <http://www.ieee802.org/15/pub/TG4b.html> (September 2006).
- [7] A. Koubãa et al., TDBS: a time division beacon scheduling mechanism for zigbee cluster-tree wireless sensor networks, *Real-Time Systems* 40 (3) (2008) 321–354. doi:10.1007/s11241-008-9063-4.
- [8] N. Abdeddaim, F. Theoleyre, M. Heusse, A. Duda, Adaptive ieee 802.15.4 mac for throughput and energy optimization, in: *International Conference on Distributed Computing in Sensor Systems (DCOSS)*, IEEE, Los Alamitos, CA, USA, 2013, pp. 223–230. doi:10.1109/DCOSS.2013.44.
- [9] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, Wirelesshart: Applying wireless technology in real-time industrial process control, in: *Real-Time and Embedded Technology and Applications Symposium, 2008. RTAS '08*. IEEE, 2008, pp. 377–386. doi:10.1109/RTAS.2008.15.
- [10] P. Thubert, T. Watteyne, R. Assimiti, An architecture for ipv6 over the tsch mode of ieee 802.15.4e, draft 8, IETF (May 2015).
- [11] Q. Wang, X. Vilajosana, 6tisch operation sublayer (6top), Tech. Rep. draft version 4, IETF (November 2015).
- [12] J.-S. Lee, Y.-W. Su, C.-C. Shen, A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi, in: *Conference of the IEEE Industrial Electronics Society (IECON)*, 2007, pp. 46–51. doi:10.1109/IECON.2007.4460126.
- [13] P. Yi, A. Iwayemi, C. Zhou, Developing zigbee deployment guideline under wifi interference for smart grid applications, *IEEE Transactions on Smart Grid* 2 (1) (2011) 110–120. doi:10.1109/TSG.2010.2091655.
- [14] T. Watteyne, A. Mehta, K. Pister, Reliability through frequency diversity: Why channel hopping makes sense, in: *Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, ACM, 2009, pp. 116–123. doi:10.1145/1641876.1641898.
- [15] Q. Lampin, D. Barthel, I. Auge-Blum, F. Valois, Exploiting long-range opportunistic links to improve delivery, delay and energy consumption in wireless sensor networks, in: *International Conference on Mobile Adhoc and Sensor Systems (MASS)*, IEEE, 2012, pp. 483–484. doi:10.1109/MASS.2012.6502556.
- [16] A. Ghosh, O. Incel, V. Kumar, B. Krishnamachari, Multi-channel scheduling algorithms for fast aggregated convergecast in sensor networks, in: *MASS*, IEEE, 2009, pp. 363–372. doi:10.1109/MOBHOC.2009.5336979.
- [17] J. N. Tsitsiklis, K. Xu, On the power of (even a little) centralization in distributed processing, in: *ACM SIGMETRICS*, 2011, pp. 161–172. doi:10.1145/1993744.1993759. URL <http://doi.acm.org.gate6.inist.fr/10.1145/1993744.1993759>
- [18] M. Yan, K.-Y. Lam, S. Han, E. Chan, Q. Chen, P. Fan, D. Chen, M. Nixon, Hypergraph-based data link layer scheduling for reliable packet delivery in wireless sensing and control networks with end-to-end delay constraints, *Information Sciences* 278 (2014) 34 – 55. doi:http://dx.doi.org/10.1016/j.ins.2014.02.006. URL <http://www.sciencedirect.com/science/article/pii/S0020025514001030>
- [19] M.R. Palattella, et al., On optimal scheduling in duty-cycled industrial iot applications using IEEE802.15.4e TSCH, *Sensors Journal, IEEE* 13 (10) (2013) 3655–3666. doi:10.1109/JSEN.2013.2266417.
- [20] M. Yigit, O. D. Incel, V. C. Gungor, On the interdependency between multi-channel scheduling and tree-based routing for WSNs in smart grid environments, *Computer Networks* 65 (0) (2014) 1 – 20. doi:10.1016/j.comnet.2014.02.025.
- [21] F. Dobsław, T. Zhang, M. Gidlund, End-to-end reliability-aware scheduling for wireless sensor networks, *IEEE Transactions on Industrial Informatics*doi:10.1109/TII.2014.2382335.
- [22] N. Accettura, M. Palattella, G. Boggia, L. Grieco, M. Dohler, Decentralized traffic aware scheduling for multi-hop low power lossy networks in the internet of things, in: *World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2013 IEEE 14th International Symposium and Workshops on a, 2013, pp. 1–6. doi:10.1109/WoWMoM.2013.6583485.
- [23] B. Pavkovic, A. Duda, W.-J. Hwang, F. Theoleyre, Efficient topology construction for RPL over IEEE 802.15.4 in wireless sensor networks, *Ad Hoc Networks* 15 (2014) 25–38. doi:http://dx.doi.org/10.1016/j.adhoc.2013.08.009. URL <http://www.sciencedirect.com/science/article/pii/S1570870513001716>
- [24] K. Srinivasan, P. Dutta, A. Tavakoli, P. Levis, Understanding the causes of packet delivery success and failure in dense wireless sensor networks, in: *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, ACM Request Permissions, 2006.

- [25] A. Cerpa, J. L. Wong, M. Potkonjak, D. Estrin, Temporal Properties of Low Power Wireless Links: Modeling and Implications on Multi-Hop Routing, in: the 6th ACM international symposium, ACM Press, New York, New York, USA, 2005, pp. 414–425.
- [26] R. Sudhaakar, P. Zand, 6tisch resource management and interaction using coap, Tech. rep., IETF (2015).
- [27] H.-S. Wang, N. Moayeri, Finite-state markov channel-a useful model for radio communication channels, Vehicular Technology, IEEE Transactions on 44 (1) (1995) 163–171. doi:10.1109/25.350282.
- [28] P. Sadeghi, R. Kennedy, P. Rapajic, R. Shams, Finite-state markov modeling of fading channels - a survey of principles and applications, Signal Processing Magazine, IEEE 25 (5) (2008) 57–80. doi:10.1109/MSP.2008.926683.
- [29] J. Tang, G. Xue, W. Zhang, Cross-layer design for end-to-end throughput and fairness enhancement in multi-channel wireless mesh networks, IEEE Transactions on Wireless Communications 6 (2007) 3482–3486.
- [30] D. Johnson, M. Yannakakis, C. H. Papadimitriou, On generating all maximal independent sets, Information Processing Letters 27 (3) (1998) 119–123.
- [31] J. C. F. Li, S. Dey, Lifetime optimization for wireless sensor networks with outage probability constraints, in: Wireless Conference 2006 - Enabling Technologies for Wireless Multimedia Communications (European Wireless), 12th European, 2006, pp. 1–7.
- [32] R. Uргаonkar, M. Neely, Delay-limited cooperative communication with reliability constraints in wireless networks, in: INFOCOM 2009, IEEE, 2009, pp. 2561–2565. doi:10.1109/INFCOM.2009.5062187.
- [33] R. Srivastava, C. Koksal, Energy optimal transmission scheduling in wireless sensor networks, Wireless Communications, IEEE Transactions on 9 (5) (2010) 1550–1560. doi:10.1109/TWC.2010.05.090275.
- [34] M. Zafer, E. Modiano, Optimal rate control for delay-constrained data transmission over a wireless channel, Information Theory, IEEE Transactions on 54 (9) (2008) 4020–4039.
- [35] T. Tran, H. Li, W. Lin, L. Liu, S. Khan, Adaptive scheduling for multicasting hard deadline constrained prioritized data via network coding, in: IEEE GLOBECOM, 2012.
- [36] W. B. Powell, Approximate Dynamic Programming: Solving the Curses of Dimensionality (Wiley Series in Probability and Statistics), Wiley-Interscience, 2007.
- [37] Forecasting with Exponential Smoothing: The State Space Approach, 2008th Edition, Springer, Berlin, 2008.
- [38] R. K.-M. Cheung, W. B. Powell, Shape - a stochastic hybrid approximation procedure for two-stage stochastic programs, Oper. Res. 48 (1) (2000) 73–79.
- [39] T. Liu, A. E. Cerpa, Talent: Temporal adaptive link estimator with no training, in: Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, SenSys '12, ACM, New York, NY, USA, 2012, pp. 253–266. doi:10.1145/2426656.2426682.
URL <http://doi.acm.org.gate6.inist.fr/10.1145/2426656.2426682>
- [40] Y. Cui, H. Wang, X. Cheng, D. Li, A. Yla-Jaaski, Dynamic scheduling for wireless data center networks, Parallel and Distributed Systems, IEEE Transactions on 24 (12) (2013) 2365–2374. doi:10.1109/TPDS.2013.5.
- [41] M. Palattella, L. Grieco, T. Watteyne, Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement.
URL <https://tools.ietf.org/html/draft-ietf-6tisch-tsch-05>