Priority-based bandwidth allocation in wireless sensor networks

Andrea Marin
Università Ca' Foscari Venezia, Italy
marin@dais.unive.it

Sabina Rossi
Università Ca' Foscari Venezia, Italy
srossi@dais.unive.it

ABSTRACT

In Wireless Sensor Networks (WSN) a set of motes monitors the environment by measuring some physical phenomena such as humidity, light, temperature, vibrations. The coexistence of different data types arises the problem of assigning the network resources in a fair way by taking into account possible different priorities among the traffic streams. In this paper we propose an allocation control scheme which is easy to implement, meets the limited resources of sensor nodes, and does not require extra control traffic in the WSN. Our scheme is based on the idea that the motes maintain a window with the classes of the latest transmissions. We propose an analytical model and provide an algorithm to study the performance of this allocation scheme with respect to the throughput and the fairness achieved. The model is a continuous time Markov chain which is proved to be dynamically reversible (although not reversible) and admits a product-form equilibrium distribution.

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of systems—*modeling techniques*; C.2.1 [**Computer-Communication Networks**]: Network architecture and design—*Wireless Communication*

Keywords

Wireless Networks; Markov Processes; Time Reversibility

1. INTRODUCTION

Wireless Sensor Networks (WSNs) are employed to collect data from an environment and send them to monitoring and control applications. Sensors are typically deployed in an environment and are responsible for sensing and measuring many physical phenomena such as temperature, humidity, pressure, radiation, air pollution levels, noise level but also more sophisticated events such as the number of people in an area or the movement of an object. Sensors themselves form a communication network which may be based on several architectures: all the nodes may implement the same wireless transmission protocol in order to form a decentralised adhoc network or there may be a hierarchical structure as in the cluster-tree topology [**?**, **?**].

It is often the case that the communication infrastructure formed by the sensors is used to transmit different types of data which may have different importance and priority. For instance in monitoring an environment the data associated with sensing vibrations may have higher priority than those measuring other phenomena such as the humidity since the former may indicate an earthquake or a tremor. Therefore, it appears natural that in the development of WSN protocols one should consider the assignment of the resources (frequency spectrum or bandwidth) according to the priorities of the sensed data. On the other hand, due to the limited amount of computational power and the limited energy supplies of the sensors, the design of protocols for the resource assignment should be as simple as possible and avoid energy loss due to heavy computations or packet collisions.

In this paper we address the problem of the bandwidth allocation among traffic classes with different priority by introducing a stateless protocol inspired by *back-CHOKe* [**?**]. The main idea of our allocation control scheme, named *Fair Allocation Control Window* (FACW), is that each sensor maintains a control window of size N that stores the traffic classes of the latest packets that have been sent by its neighbours (i.e., the other sensors it can listen to) or by itself. Each traffic class c can be present in the window at most h_c times, where higher values of h_c correspond to higher traffic priority. When a sensor collects data with type c, it behaves as follows: if the number of c-classes in its window is less then h_c then it transmits the packet, otherwise it waits a random time (or drop the packet) and retry later. If the packet is sent then the window is updated by inserting an object of class c according to a First-In-First-Out (FIFO) policy. We study the behaviour of this protocol under different scenarios. We propose a performance evaluation of the FACW protocol based on the assumption that the data packets are collected by the sensors according to independent Poisson processes. We present an exact model based on the analysis of a Continuous Time Markov Chain (CTMC) with finite state space. We show that the model is dynamically reversible [**?**, **?**, **?**, **?**], although not reversible, and give a closed form expression for its invariant measure. The stationary performance indices are expressed as functions of the normalising constant which is derived algorithmically according to a convolution algorithm. The availability of a numerically efficient approach for the performance evaluation allows for the parameterisation of a WSN without resorting to computationally expensive simulations.

Structure of the paper. In Section **??** we describe our bandwidth allocation control scheme. In Section **??** we present the performance evaluation of the protocol and give the algorithm for the computation of the performance indices. Section **??** shows our scheme at work under various scenarios. Section **??** discusses some related work. Finally, Section **??** concludes the paper.

2. PROTOCOL DESIGN

In this section we first introduce the design goals of the FACW protocol and then we describe how it works.

2.1 Design goals

The design of FACW aims at satisfying the following goals:

1. *Stateless architecture.* Due to the limited physical resources in the motes, we aim at reducing the computational cost and the memory usage at each mote.
2. *Localized behaviour.* The protocol activities are based on local information and do not affect the whole system.
3. *Avoid transmissions of extra packets.* Control packet transmission should be avoided in order to reduce the energy dissipation of the system
4. *Fair bandwidth allocation among different traffic flows with the same priority.* Our protocol aims at satisfying the *max-min* fairness criterion among different traffic flows with the same priority. We will discuss this objective in more details in Section **??**, but intuitively we do not want that a flow with low requirements is slowed down while there exists another flow with the same priority which is using more resources [**?**, **?**].
5. *Flexible regulation of traffic priorities.* The allocation of the bandwidth follows a soft-priority based scheme in order to prevent lower priority traffic flows to starve because of the presence of a greedy higher traffic flow.
6. *Easiness of implementation.* The protocol must be easy to implement within the actual motes software.

2.2 FACW

The main idea of FACW is that data traffic in a WSN can be classified into a finite set of M classes $\mathcal{K} = \{c_1, c_2, \dots, c_M\}$. Each mote maintains a control window of size N in which the classes of the latest N transmissions (listened or performed) are stored. In the window, at most h_c entries of class c can appear. We stress the fact that the window stores only the class identifier of a transmission and not the sent packet. So, if we assume a practical situation with 16 classes, each class can be encoded by 4 bits and hence a window can be stored in few bytes. In case the mote generates a packet of class c when in its window there are already h_c entries of class c, the packet is rescheduled for transmission after a back-off time or is simply dropped. Otherwise, in case of generation of a class c packet and the number of c-entries in the window is strictly lower than h_c, then the packet is sent and the window is updated according to a FIFO policy. It should be clear that larger window sizes imply a lower bandwidth usage, whereas lower values of N make the transmission more aggressive. The role of h_c is that of modelling class priority. Allowing more entries of a class c in the control window reduces the probability of c-packet dropping/delaying and hence its priority is larger than that of a traffic class d with $h_d < h_c$. The initialisation of the window is arbitrary. If necessary, we can assume that there exists a class c of data traffic (e.g., the packets used for controlling the routing) that has $h_c = N$ and whose rate is slow. The presence of this class ensures that the starvation of all the other traffic classes never occurs because of the control window.

In the following sections we present a numerically tractable model that can be used to parametrise the protocol, i.e., decide the window size and the values of h_c for $c \in \mathcal{K}$.

3. PERFORMANCE EVALUATION

In this section we present a CTMC model of the FACW protocol. The numerical tractability of this protocol allows us to use it to set the parameters in the protocol implementation, i.e., the window size N and the values for the threshold h_c for each traffic class $c \in \mathcal{K}$. The model considers a single window and is subject to the following assumptions:

- Packets are generated according to independent Poisson processes whose rates may depend on the window state. This allows us to model situations in which the mote modulates its harvesting rate according to the population of the control window. We can also deal with the case in which the class c packets which are not sent are delayed and hence the packet generation rate is increased because the sensor data production rate is summed to the packet retransmission rate.
- We consider a network topology in which every mote senses the transmission of every other mote. This is a common assumption in tree-structured WSNs in which it is assumed that all the motes with the same parent interfere in their transmissions because they are relatively geographically close. This requirement is needed because here we aim to study the network performance and hence we assume that all the nodes share the same contention window. Nevertheless, if we are interested in the analysis of the performance of a single mote, this requirement is not needed.

3.1 CTMC model

We consider a set $\mathcal{K} = \{c_1, c_2, \dots, c_M\}$ of M distinct traffic classes and assume that each node maintains a window $\mathcal{W}$ of size N storing the transmission classes of the most recent sensed data according to a FIFO policy. An arrival can be due to a sensor data harvesting or to a listening to another node transmission. We denote the state of the window by $\vec{x} = (x_1, x_2, \dots, x_N)$, where $x_i \in \mathcal{K}$, and let $|\vec{x}|_c = \sum_{i=1}^{N} \delta_{x_i=c}$ be the total number of occurrences of class c in $\mathcal{W}$. We assume that data of different traffic classes are generated according to independent Poisson processes whose rates $\lambda_c(j)$, with $c \in \mathcal{K}$ and $1 \leq j \leq N$, depend on the number of objects $j = |\vec{x}|_c$ of class c that are present in the window. Clearly, the process $X(t)$ that describes the state of $\mathcal{W}$ is a homogeneous continuous time Markov chain (CTMC) with finite state space. In the window there can be at most h_c objects of class c, with $c \in \mathcal{K}$. If $h_c = N$ then there is no constraint on the maximum number of objects of the same class in the window. Let $\vec{x} = (x_1, \dots, x_N)$ be the state of the control window, then the transition rates in the

CTMC infinitesimal generator are: for $\vec{x} \neq \vec{x}'$,

$$q(\vec{x}, \vec{x}') = \begin{cases} \lambda_c(|\vec{x}|_c) & \text{if } \vec{x}' = (c, x_1, \dots, x_{N-1}) \\ & \text{and } |\vec{x}|_c < h_c \\ 0 & \text{otherwise.} \end{cases}$$

3.2 Closed form stationary distribution

We derive the stationary distribution of process $X(t)$. The state space of $X(t)$ is $\mathcal{S} = \{\vec{x} \in \mathcal{K}^N : |\vec{x}|_c \leq h_c \text{ for all } c \in \mathcal{K}\}$. Note that the state space of $X(t)$ is finite and its transition graph is irreducible. Hence the CTMC has a unique limiting distribution independent of its initial state.

Theorem 1. *The stationary distribution $\pi(\vec{x})$ of $X(t)$ for the FIFO policy is given by the following expression:*

$$\pi(\vec{x}) = \frac{1}{G} \prod_{c \in \mathcal{K}} \prod_{j=0}^{|\vec{x}|_c - 1} \lambda_c(j)\,, \qquad (1)$$

where $G = \sum_{\vec{x} \in \mathcal{S}} \prod_{c \in \mathcal{K}} \prod_{j=0}^{|\vec{x}|_c - 1} \lambda_c(j)$.

The proof of Theorem **??** is based on the notion of *dynamic reversibility* [**?**, **?**, **?**, **?**] which generalises the well-known concept of reversibility by considering those CTMCs which are stochastically identical to their reversed process modulo a state renaming ϱ. More formally, if ϱ is an involution over the state space $\mathcal{S}$ of a CTMC $X(t)$, $X(t)$ is dynamically reversible with respect to ϱ if it is stochastically identical to its reversed process where the state names are changed according to ϱ. Dynamically reversible Markov chains are characterized by a set of detailed balance equations expressed in terms of the steady-state distribution π and the transition rates q_{ij}, for $i, j \in \mathcal{S}$, of the Markov process.

Proposition 1. *A stationary CTMC with state space $\mathcal{S}$ is dynamically reversible w.r.t. a renaming ϱ on $\mathcal{S}$ if and only if there exists a set of positive real numbers π_i summing to unity, with $i \in \mathcal{S}$, such that the following system of detailed balance equations are satisfied: for $i, j \in \mathcal{S}$, $i \neq j$:*

$$\pi_i q_{ij} = \pi_j q_{\varrho(j)\varrho(i)}$$

and $q_i = q_{\varrho(i)}$. If such a solution π_i exists then it is the stationary distribution of $X(t)$.

The steady-state distribution of a dynamically reversible CTMC can be expressed in terms of the transition rates.

Proposition 2. *Let $X(t)$ be a stationary CTMC with state space $\mathcal{S}$ which is dynamically reversible with respect to a renaming ϱ over $\mathcal{S}$. Let $i_0 \in \mathcal{S}$ be and arbitrary reference state. Let $i \in \mathcal{S}$ and $i = i_n \to i_{n-1} \to \cdots \to i_1 \to i_0$ be a chain of one-step transitions. Then, for $C_{i_0} \in \mathbb{R}^+$,*

$$\pi_i = C_{i_0} \prod_{k=1}^{n} \frac{q_{\varrho(i_{k-1})\varrho(i_k)}}{q_{i_k i_{k-1}}}\,. \qquad (2)$$

Proof of Theorem **??**. The proof is structured as follows: we first make a claim that $X(t)$ is dynamically reversible and then derive Expression (**??**) of the stationary distribution. Finally, by using Proposition **??** we prove the claim.

Claim 1. *The process $X(t)$ for the FIFO policy is dynamically reversible w.r.t. the renaming ϱ on $\mathcal{S}$ defined by $\varrho(\vec{x}) = \vec{x}^R$ where $\vec{x} = (x_1, \dots, x_N)$ and $\vec{x}^R = (x_N, \dots, x_1)$.*

We assume that $h_c = N$ for all the traffic classes $c \in \mathcal{K}$. We will see later that this assumption does not limit the validity of this proof. Assuming Claim **??** we use Proposition **??** to derive the expression of the stationary distribution π. Let us take a reference state $\vec{x}_0 = \vec{c_1}^N$ the N-sized vector whose entries are all equal to c_1, and let us derive the stationary probability of a general state $\vec{x} \in \mathcal{S}$. Consider the sequence of arrivals that starting from state $\vec{x}$ take the model to state $\vec{x}_0$ consisting in the arrival of exactly N objects of class c_1. We denote this path as follows:

$$\vec{x}_1 \equiv \vec{x} \xrightarrow{c_1} \vec{x}_2 \xrightarrow{c_1} \vec{x}_3 \cdots \xrightarrow{c_1} \vec{x}_{N+1} \equiv \vec{x}_0\,,$$

where we have labelled the arrows with the arriving classes. Notice that the reversed path from $\vec{x}_0^R = \vec{c_1}^N$ to $\vec{x}^R = (x_N, \dots, x_1)$ exists in the same process and is formed by the arrival of the sequence of traffic classes x_1, x_2, $\dots x_N$. Suppose that $\vec{x}$ has $K \leq N$ objects of class c_1 in positions $i_1 < i_2 < .. < i_K \leq N$. The product of the rates in the forward path must take into account that the number of c_1 in the window starts from K and keeps increasing a unity at each arrival with the exception of the case in which an object of class c_1 is discarded. The c_1 in position i_k will be in position N after $N - i_k$ arrivals. The arrival $N - i_k + 1$ will leave the same number of class c_1 objects in the queue which is $N - i_k + 1$ due to the arrival plus the $k - 1$ which are with index lower than i_k. Therefore, the product of the rates of the forward path is:

$$\prod_{j=K}^{N-1} \lambda_{c_1}(j) \cdot \prod_{k=1}^{K} \lambda_{c_1}(N - i_k + k)\,. \qquad (3)$$

The product of the rates in the reversed path is:

$$\left(\prod_{c \in \mathcal{K} \smallsetminus \{c_1\}} \prod_{j=0}^{|\vec{x}|_c - 1} \lambda_c(j) \right) \cdot \prod_{k=1}^{K} \lambda_{c_1}(N - i_k + k)\,, \qquad (4)$$

where the first factor is due to the arrivals of class $c \neq c_1$ objects while the second is due to class c_1 object arrival. Indeed if in $\vec{x}$ the objects of class c_1 are present in position i_k, $1 \leq k \leq K$, this means that the i_k-th arrival will be a c_1. The number of occurrences of c_1 in the window is $N - i_k + 1$ (due to the previus $i_k - 1$ arrivals) plus $k - 1$ due to the c_1 objects already arrived. Using Proposition **??**, by Equations (**??**) and (**??**), we can derive $\pi(\vec{x})$:

$$\pi(\vec{x}) = \pi(\vec{x}_0) \frac{\prod_{c \in \mathcal{K} \smallsetminus \{c_1\}} \prod_{j=0}^{|\vec{x}|_c - 1} \lambda_c(j)}{\prod_{j=K}^{N-1} \lambda_{c_1}(j)}\,. \qquad (5)$$

This says that if Claim **??** is true, then Equation (**??**) is the stationary distribution of $X(t)$. Indeed, by Equation (**??**):

$$\pi(\vec{x}_0) = \frac{1}{G} \prod_{j=0}^{N-1} \lambda_{c_1}(j)$$

and then by Equation (**??**) we can write

$$\pi(\vec{x}) = \frac{1}{G} \frac{\prod_{j=0}^{N-1} \lambda_{c_1}(j) \prod_{c \in \mathcal{K} \smallsetminus \{c_1\}} \prod_{j=0}^{|\vec{x}|_c - 1} \lambda_c(j)}{\prod_{j=K}^{N-1} \lambda_{c_1}(j)}$$

that is equal to

$$\pi(\vec{x}) = \frac{1}{G} \prod_{j=0}^{K-1} \lambda_{c_1}(j) \prod_{c \in \mathcal{K} \smallsetminus \{c_1\}} \prod_{j=0}^{|\vec{x}|_c - 1} \lambda_c(j) \,.$$

Now since $|\vec{x}|_{c_1} - 1 = K$, we obtain

$$\pi(\vec{x}) = \frac{1}{G} \prod_{c \in \mathcal{K}} \prod_{j=0}^{|\vec{x}|_c - 1} \lambda_c(j)$$

proving that Equation (**??**) is indeed the stationary distribution of $X(t)$.
Let us now prove that Claim **??** is true by using Proposition **??**. We show that Equation (**??**) satisfies the detailed balance equations for the following models:

1. FIFO with $h_c = N$ for all $c \in \mathcal{K}$, which is the case we used to derive the candidate expression,
2. FIFO with arbitrary $1 \leq h_c < N$ for some $c \in \mathcal{K}$.

Let $\vec{x} \in \mathcal{S}$ be $(x_1, \dots x_N)$. We distinguish two cases.
Case 1. The detailed balance equation becomes:

$$\pi(\vec{x})\lambda_d(|\vec{x}|_d) = \pi(d, x_1, \dots, x_{N-1})\lambda_{x_N}(|\vec{x}|_{x_N} - \delta_{x_N \neq d}) \,.$$

If $x_N \neq d$, by substituting the expression (**??**) of π we obtain:

$$\prod_{c \in \mathcal{K}} \prod_{j=0}^{|\vec{x}|_c - 1} \lambda_c(j) \lambda_d(|\vec{x}|_d) =$$
$$\prod_{c \in \mathcal{K}} \prod_{j=0}^{|\vec{x}|_c - 1} \lambda_c(j) \frac{\lambda_d(|\vec{x}|_d)}{\lambda_{x_N}(|\vec{x}|_{x_N} - 1)} \lambda_{x_N}(|\vec{x}|_{x_N} - 1) \,,$$

which is an identity. If $x_N = d$ the detailed balance equation is trivially an identity since $\pi(\vec{x}) = \pi(x_N, x_1, \dots, x_{N-1})$ and also the transition rates are identical.
Case 2. Let us consider now the case of $1 \leq h_c < N$ for some c. Observe that if there exists a transition from $\vec{x}$ to a different state due to the arrival of a class d, then $|\vec{x}|_d < h_d$. If $x_N = d$, this implies that also the reversed transition is possible and we already showed that the detailed balance equation is satisfied. If $x_N \neq d$, then clearly $|\vec{x}|_{x_N} \leq h_{x_N}$ that implies that the reversed transition is allowed since it occurs in a state with $|\vec{x}|_{x_N} - 1$ objects of class x_N. □

In most practical applications, we are interested in knowing the stationary probability of observing a state in which the occurrences of each class $c_1, \dots, c_M$ are $n_{c_1}, \dots, n_{c_M}$ whatever is their order. Corollary **??** provides an analytical expression for such an aggregated equilibrium probability.

Corollary 1. *Let* $\mathbf{n} = (n_{c_1}, \dots, n_{c_M})$ *with* $0 \leq n_c \leq h_c$ *for all* $c \in \mathcal{K}$ *and* $\sum_{c \in \mathcal{K}} n_c = N$. *The stationary probability of observing the aggregated state with* n_c *elements of class* c *for all* $c \in \mathcal{K}$ *is:*

$$\pi_A(\mathbf{n}) = \frac{1}{G} \binom{N}{n_{c_1}, n_{c_2}, \dots, n_{c_M}} \prod_{c \in \mathcal{K}} \prod_{j=0}^{n_c - 1} \lambda_c(j) \,,$$

where $\mathbf{n}$ *belongs to the set of aggregated states*

$$\mathcal{S}_{\mathcal{K},N} = \{\mathbf{n} : \sum_{c \in \mathcal{K}} n_c = N \text{ and } 0 \leq n_c \leq h_c \ \forall c \in \mathcal{K}\} \,.$$

We denote the normalising constant and the aggregated stationary distribution of the system model consisting of a set of traffic classes $\mathcal{K}$ and a window size N as $G_{\mathcal{K},N}$ and $\pi_{\mathcal{K},N}$, respectively. The marginal equilibrium distribution for each class is given by Lemma **??**.

Lemma 1. *The marginal stationary probability of observing exactly* δ *objects of class* $d \in \mathcal{K}$ *in the window, with* $0 \leq \delta \leq h_d$*, is:*

$$\pi^d_{\mathcal{K},N}(\delta) = \binom{N}{\delta} \left(\prod_{j=0}^{\delta-1} \lambda_d(j) \right) \frac{G_{\mathcal{K} \smallsetminus \{d\}, N-\delta}}{G_{\mathcal{K},N}} \,.$$

3.3 Performance indices

In this section we introduce a set of performance indices and show how to compute them efficiently.

Definition 1 (Admission rate). *The admission rate for a class* $c \in \mathcal{K}$ *is the rate associated with the event of transition from a state* $\vec{x}$ *with* $|\vec{x}|_c = 0$ *to a state* $\vec{x}'$ *with* $|\vec{x}'|_c = 1$ *when the model is in steady-state. The global admission rate is the sum of the admission rates for each* $c \in \mathcal{K}$.

Definition 2 (Rejection rate). *The rejection rate for a traffic class* $c \in \mathcal{K}$ *is the rate associated with the event of rejecting the arrival of class* c *because the number of objects of class* c *in the window is* h_c. *The global rejection rate is the sum of the rejection rates for each traffic class* $c \in \mathcal{K}$.

The admission rate for a specific traffic class and the global admission rate can be computed as in Corollary **??**, while the rejection rate for a specific traffic class and the global rejection rate can be computed as in Corollary **??**.

Corollary 2. *In steady-state, the admission rate for a traffic class* $d \in \mathcal{K}$ *is:*

$$X^d_{\mathcal{K},N} = \lambda_d(0) \frac{G_{\mathcal{K} \smallsetminus \{d\}, N}}{G_{\mathcal{K},N}} \,, \qquad (6)$$

and the global admission rate is:

$$X_{\mathcal{K},N} = \sum_{c \in \mathcal{K}} \lambda_c(0) \frac{G_{\mathcal{K} \smallsetminus \{c\}, N}}{G_{\mathcal{K},N}} \,, \qquad (7)$$

Proof. The proof follows straightforwardly from Lemma **??** by observing that $X^d_{\mathcal{K},N} = \lambda_d(0)\pi^d_{\mathcal{K},N}(0)$. □

Corollary 3. *In steady-state, the rejection rate for a traffic class* $d \in \mathcal{K}$ *is:*

$$Y^d_{\mathcal{K},N} = \lambda_d(h_d) \binom{N}{h_d} \prod_{j=0}^{h_d - 1} \lambda_d(j) \frac{G_{\mathcal{K} \smallsetminus \{d\}, N-h_d}}{G_{\mathcal{K},N}} \,, \qquad (8)$$

and the global rejection rate is:

$$Y_{\mathcal{K},N} = \sum_{c \in \mathcal{K}} \lambda_c(h_c) \binom{N}{h_c} \prod_{j=0}^{h_c - 1} \lambda_c(j) \frac{G_{\mathcal{K} \smallsetminus \{c\}, N-h_c}}{G_{\mathcal{K},N}} \,. \qquad (9)$$

PROOF. The proof follows straightforwardly from Lemma **??** by observing that $Y^d_{\mathcal{K},N} = \lambda_d(h_d)\pi^d_{\mathcal{K},N}(h_d)$. □

By applying Lemma **??** we can compute the expected number of objects of a given traffic class in the window when the model is in steady-state and the throughput for each class.

COROLLARY 4. *In steady-state, the expected number of objects of class $d \in \mathcal{K}$ in the window is:*

$$\overline{N}^d_{\mathcal{K},N} = \sum_{\delta=1}^{h_d} \delta \binom{N}{\delta} \prod_{j=0}^{\delta-1} \frac{G_{\mathcal{K}\setminus\{d\},N-\delta}}{G_{\mathcal{K},N}} \,. \qquad (10)$$

COROLLARY 5. *In steady-state the throughput for a traffic class $d \in \mathcal{K}$ is:*

$$\lambda^*_d = \sum_{\delta=0}^{h_d-1} \lambda_d(\delta) \binom{N}{\delta} \left(\prod_{j=0}^{\delta-1} \lambda_d(j) \right) \frac{G_{\mathcal{K}\setminus\{d\},N-\delta}}{G_{\mathcal{K},N}} \,. \qquad (11)$$

Finally, we introduce an index to measure the fairness of the bandwidth allocation among the set of traffic classes.

DEFINITION 3. *Let $\mathcal{K}_1 \subseteq \mathcal{K}$ be a subset of the traffic classes whose elements have the same priority. Assume that the arrival rates for the classes in $\mathcal{K}_1$ are independent of the state of the window, i.e., for any $c \in \mathcal{K}_1$, $\lambda_c(j) = \lambda_c$ for all $0 \le j \le h_c$. The fairness index Φ_{cd} of a class c with respect to a class d with $c, d \in \mathcal{K}_1$ is defined as follows:*

$$\Phi_{cd} = \min\left(\lambda_c - \lambda^*_c, \max(\lambda^*_d - \lambda^*_c, 0)\right) .$$

The global fairness index for $\mathcal{K}_1$ is defined as:

$$\Phi_{\mathcal{K}_1} = \sum_{c \in \mathcal{K}_1} \frac{\lambda_c}{\lambda_{\mathcal{K}_1}} \sum_{d \in \mathcal{K}_1} \Phi_{cd} \,,$$

where $\lambda_{\mathcal{K}_1} = \sum_{c \in \mathcal{K}_1} \lambda_c$.

The next proposition states that when the fairness index is 0 we achieve the *max-min* fairness, i.e., a flow with low requirements is never slowed down while there exists another flow with the same priority which is using more resources [**?**].

PROPOSITION 3. *Let $\mathcal{K}_1 \subseteq \mathcal{K}$. The fairness index $\Phi_{\mathcal{K}_1}$ is 0 if and only if the allocation of the bandwidth $\lambda^*_{\mathcal{K}_1} = \sum_{c \in \mathcal{K}_1} \lambda^*_c$ is max-min fair.*

3.4 Computation of the normalising constant

The expression for the normalising constant given by Theorem **??** is computationally expensive and prone to numerical instability problems. In this section we provide an efficient algorithm for computing the normalising constant based on its convolution property. We define $\tau_{\mathcal{K}} \in \mathbb{N}$ as the maximum number of slots that the traffic classes in $\mathcal{K}$ would occupy in an infinite size window, i.e.,

$$\tau_{\mathcal{K}} = \sum_{c \in \mathcal{K}} h_c \,.$$

Notice that, given a partition $\mathcal{K}_1$ and $\mathcal{K}_2$ of the set of traffic classes $\mathcal{K}$, it clearly holds that $\tau_{\mathcal{K}} = \tau_{\mathcal{K}_1} + \tau_{\mathcal{K}_2}$.

LEMMA 2. *Let $\mathcal{K}$ be the set of traffic classes and let $\mathcal{K}_1$ and $\mathcal{K}_2$ be a partition of $\mathcal{K}$. Then, the normalising constant can be defined by the following recursive relation:*

$$G_{\mathcal{K},N} = \sum_{j=\max(0,N-\tau_{\mathcal{K}_1})}^{\min(N,\tau_{\mathcal{K}_2})} \binom{N}{j} G_{\mathcal{K}_1,N-j} G_{\mathcal{K}_2,j} \,. \qquad (12)$$

Let us order the traffic classes $c_1, \ldots c_M \in \mathcal{K}$ and let $\vec{h} = (h_{c_1}, \ldots, h_{c_M})$. We compute the normalising constant as shown in Algorithm **??** where we use the convention that array positions start from 1 and empty products have value 1. Notice that if $\mathcal{K}$ is a singleton, then the normalising constant $G_{\mathcal{K},\min(N,h_c)}$ can be computed easily as:

$$G_{\{c\},min(N,h_c)} = \prod_{j=0}^{\min(N,h_c)-1} \lambda_c(j) \,. \qquad (13)$$

Algorithm 1: Convolution algorithm

```
input  : $\mathcal{K}, \vec{h}, N$
output: $G_{\mathcal{K},N}$
prevcol ← $[0, \ldots, N]$;
newcol ← $[0, \ldots, N]$;
{Initialise the first column};
for $i \leftarrow 0$ to $\min(N, \tau_{\{c_1\}})$ do
    newcol$(i+1) \leftarrow \prod_{j=0}^{i-1} \lambda_{c_1}(j)$;
end
for $d \leftarrow 2$ to $M$ do
    prevcol ← newcol;
    newcol$(1) \leftarrow 1$;
    $\mathcal{K}_1 \leftarrow \{c_1, \ldots, c_d\}$;
    {Compute $G_{\mathcal{K}_1,i}$ for $i = 1, \ldots \min(N, \tau_{\mathcal{K}_1})$ and store the
    result in newcol$(i+1)$};
    for $i \leftarrow 1$ to $\min(N, \tau_{\mathcal{K}_1})$ do
        newcol$(i+1) \leftarrow 0$;
        for $j \leftarrow \max(0, i - \tau_{\mathcal{K}_1 \setminus \{c_d\}})$ to $\min(N, \tau_{\mathcal{K}_1})$ do
            newcol$(i+1) \leftarrow$ newcol$(i+1)$
                $+\binom{i}{j}$prevcol$(i-j+1) \prod_{z=0}^{j-1} \lambda_d(z)$;
        end
    end
end
$G_{\mathcal{K},N} \leftarrow$ newcol$(N+1)$;
```

4. APPLICATION

We study different configurations of FACW under two scenarios consisting of 20 classes of traffic types. In the first case (S1) the total packet generation rate is 58.45 with a standard deviation of 4.3364, while in the second (S2) we have the same total generation rate with a standard deviation of 1.3516. The arrival rates for $S1$ and $S2$ are shown in Table **??**. In the following we assume $\lambda_c(n_c) = \lambda_c$ for all classes $c \in \mathcal{K}$ and $0 \le n_c < h_c$.

4.1 Impact of window size on admitted flow

In this section we study the impact of the window size on the admitted flow and on the fairness index for the scenarios S1 and S2 assuming $h_c = 1, 2$ for all $c \in \mathcal{K}$ (see Figures **??** and **??**). Plots (a) show the total throughput with respect

Class c	λ_c $(S1)$	λ_c $(S2)$
1	1.00	1.20
2	1.30	2.30
3	1.50	1.50
4	1.80	2.00
5	3.80	3.80
6	1.20	2.40
7	1.50	2.20
8	1.72	3.30
9	1.12	2.62
10	8.00	3.00
11	1.00	3.21
12	1.30	2.25
13	1.35	4.35
14	6.78	5.00
15	4.10	4.10
16	1.20	1.64
17	1.66	1.66
18	1.70	2.70
19	1.44	2.44
20	20.0	6.78

Table 1: Arrival rates for scenarios S1 and S2

to a given window size, while plots (b) of these figures show the admitted rates for the slowest and the fastest classes as functions of the window size. We notice that, as desired, the fastest classes slow down much more quickly than those that are less aggressive, thus improving the fairness in the resource allocation. Plots (c) of Figures **??** and **??** show the fairness index. Notice that total fairness is achieved when we admit almost all the streams (window size 1) or when we block them all. Nevertheless, the absolute value is always below 1.4 in S1 and 2.5 in S2 which are reasonable low values. We will study in more details in Section **??** the impact of h_c and of the window size on the fairness index.

4.2 Impact of h_c on the fairness index

In this section we consider the scenario S1 and we assume $T = 25$. We configure the window size such that λ^* is maximum under the constraint $\lambda^* < T$. We study the system for $h_c = 1, \ldots, 9$ with $c \in \mathcal{K}$. The results are shown in Table **??** and the plot of $\Phi_{\mathcal{K}}$ as function of h_c is shown in Figure **??**. We can see that the fairness is improved by larger values of h_c but, in order to control the maximum throughput, it requires larger windows and more memory. Hence, a trade-off between memory occupancy and desired fairness arises.

4.3 Different priority traffic streams

In this part we study a scenario in which we consider two pairs of streams $1, 2$ and $3, 4$ in a setting with the following packet generation rates

$$(6.0, 6.0, 18.0, 18.0, 3.8, 1.2, 1.5, 1.72, 1.12, 8.0)\,.$$

Streams 1 and 2 (resp. 3 and 4) have the same rates but the priority of 2 (resp. 4) is higher than that of 1 (resp. 3). We model this by setting $h_1 = h_3 = 1$ and $h_2 = h_4 = 3$. In Figure **??** we show the throughput of the four streams together with the total throughput. We notice that while with the increasing of the control window's size the total throughput obviously decreases, the reduction of the bandwidth assigned to the high priority traffic streams 2 and 4 is much slower than that experienced by the lower priority streams 1 and 3.

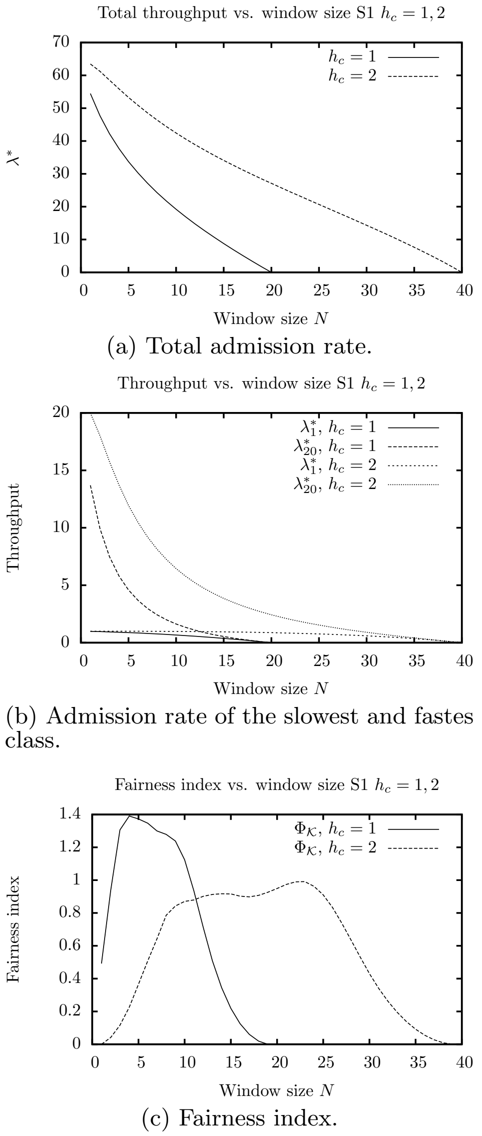
h_c	N	λ^*	Φ
1	8	24.2832	1.2786
2	22	24.5006	0.9897
3	38	24.1746	0.8513
4	54	24.5124	0.7537
5	70	24.9674	0.6807
6	87	24.9928	0.6249
7	105	24.7372	0.5800
8	123	24.6059	0.5430
9	140	24.8422	0.5117

Table 2: Impact of h_c on the fairness index

5. RELATED WORK

We discuss the works related to our contribution in two steps. First, we compare our approach with other works which address the problem of congestion control in WSNs (with or without priorities). Secondly, we compare our theoretical contribution in terms of CTMC analysis with respect to the literature.

The problem of bandwidth assignment in wireless sensor networks have been addressed by a large number of papers (see [**?**] and the references therein). In [**?**] the authors introduce the Adaptive Rate Control (ARC)scheme. In this scheme each mote estimates the number of downstreams and the bandwidth is split proportionally among the local and router through traffic. The solutions proposed in [**?**, **?**] use control packets to avoid congestion. Another important contribution is the rate control scheme introduced in [**?**] where the authors consider a tree-structured network and each mote estimates the average rate at which packets can be sent and divide it by the number of children motes downstream obtaining the maximum flow associated with each child mote. In [**?**] the authors present the SPEED protocol to achieve soft real time communication in WSNs. SPEED exploits little knowledge about the network and provides a mechanism for the packet routing that allows for a fair delivery time for the data packets. All these congestion control schemes do not consider traffic priorities, and require the transmission of control packets (which may be done by piggybacking). In [**?**] the authors propose a probabilistic approach to control the bandwidth assignment in WSNs based on the IEEE 802.15.4 standard in a tree-structured WSN. The protocol aims at obtaining the fairness among the nodes rather than among the traffic types. Traffic priorities are considered in [**?**, **?**] that exploit earliest deadline first scheduling and priority queues, respectively. The protocols proposed in [**?**] require an accurate knowledge of the network topology and divide the motes into cell. Then, the communication intra-cell and extra-cell are handled in different ways. The scheme proposed in [**?**] simply proposes a priority scheduler for each mote. Although this allows a single node to use priority for its own transmission, the correlation among the motes is not taken into account. From the point of view of the stochastic analysis, the most related works are those presented in [**?**, **?**]. In [**?**] the authors introduce CHOKe, i.e., a congestion control mechanism that allows the approx-



(a) Total admission rate.

(b) Admission rate of the slowest and fastes class.

(c) Fairness index.

Figure 1: Performance in Scenario 1 with $h_c = 1, 2$

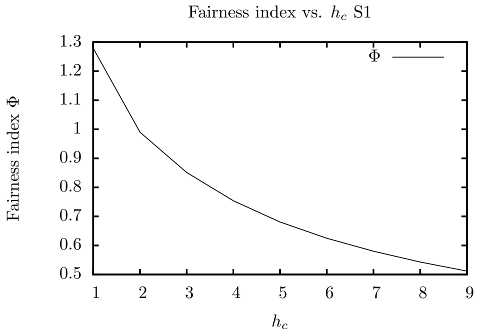
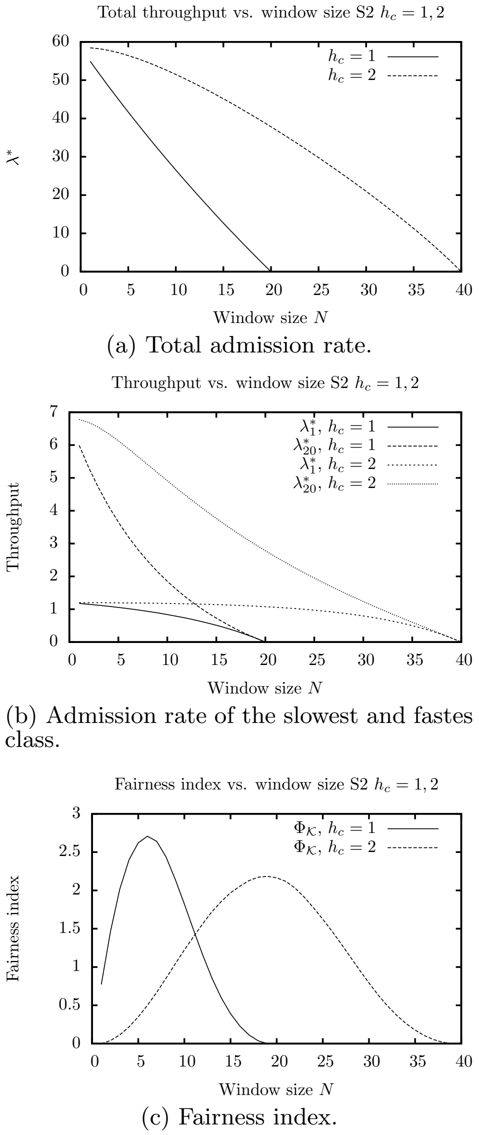


Figure 2: Fairness as function of h_c



(a) Total admission rate.

(b) Admission rate of the slowest and fastes class.

(c) Fairness index.

Figure 3: Performance in Scenario 2 with $h_c = 1, 2$

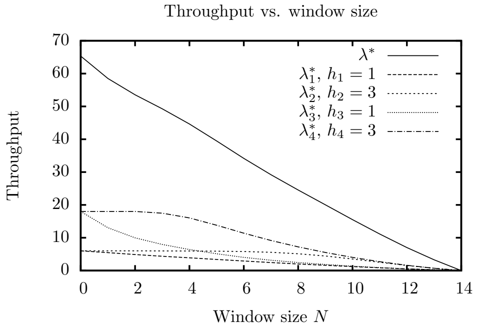


Figure 4: Throughput of classes with same rate but different priority.

imate fair sharing of a bandwidth among a set of competing customers. The authors propose a model for the analysis of *back-CHOKe* which is based on maintaining a window with the latest N packets arrived at the bottleneck. The packets coming from a source which is present in the window are discarded. Packet arrivals occur according to independent homogeneous Poisson processes, i.e., the model is a continuous time version of King's model for the FIFO cache under the Independence Reference Model assumption (IRM) [**?**]. With respect to these papers, we propose a more sophisticated model in which the window may contain a number of replicas that depends on the traffic type. Moreover, we give an efficient algorithm to compute the performance measures that implements a convolution on the finite state space of the CTMC. Indeed, the algorithm developed in [**?**] is not applicable to our model due to the possible presence of duplicated items in the window. Finally, with respect to the models studied in [**?**, **?**, **?**], we relax the requirements of the IRM by allowing the rate of the Poisson processes generating the data at the motes to depend on the window state.

6. CONCLUSION

In this paper we have proposed an allocation control scheme, named FACW, for the bandwidth assignment in WSNs. The FACW protocol is easy to implement, consumes few resources in the motes and does not require extra control traffic in the WSN. We showed that it is able to handle traffic streams with different priorities and can reach a good level of fairness among streams with the same priority. Its main idea consists in maintaining at each mote a window with the latest traffic types perceived and dropping packets of the types that have reached their maximum population in the window. Under the assumption of Poisson generated traffic, we have proposed a model which is analytically tractable and gave an algorithm to efficiently derive the performance indices. The model extends previous ones such as those developed for studying FIFO caches [**?**] and back-CHOKe [**?**] in two directions: it allows for multiple entries of the same object in the window (allowing to control the priority of the traffic streams), and it considers that the traffic generation rate may depend on the state of the window at a certain epoch. Future works include providing an implementation of the protocol and performing simulations to assess the performance under more general scenarios.