# Connection-Oriented Communication Protocol in Wireless Sensor Networks

Giuseppe Amato[1], Paolo Baronti[1], Stefano Chessa[1,2],
[1]Istituto di Scienza e Tecnologie dell'Informazione,
Area della Ricerca CNR di Pisa, Via G. Moruzzi 1, 56124 Pisa, Italy.

[2]Dipartimento di Informatica, Universitá di Pisa,
Via Buonarroti 2, 56127 Pisa,Italy.

## Abstract

Several sensor network applications based on data diffusion and data management can determine the communication transfer rate between two sensors beforehand. In this framework, we consider the problem of energy efficient communication among nodes of a wireless sensor network and propose an application-driven approach that minimizes radio activity intervals and prolongs network lifetime. On the basis of possible communication delays we estimate packet arrival intervals at any intermediate hop of a fixed-rate data path. We propose two different strategies of radio activity minimization that maintain the radio switched on just in the expected packet arrival intervals and guarantee low communication latency. We define a probabilistic model that allows the evaluation of the packet loss probability that results from the reduced radio activity. The model is used to optimally choose the radio activity intervals that achieve a certain probability of successful packet delivery. Relying on the probabilistic model we also define a cost model that estimates the energy consumption of the proposed strategies, under specific settings. We finally validate our work with a simulation made with Tossim (the Berkeley motes' simulator). The simulation results confirm the validity of the approach and the accuracy of the analytic models.

**Keywords.** Wireless sensor network, energy efficiency, communication channel, constant data rate

## 1 Introduction

The need for flexible and efficient environmental monitoring led to the recent proliferation of research on wireless sensor networks [1]. Wireless sensors are low cost microsystems equipped with sensing devices and a wireless network interface. Sensors are spread in a sensing field without any predetermined infrastructure and cooperate to execute common monitoring tasks.

Due to the lack of communication infrastructure and the limited range of wireless communication, the sensor network is generally multihop. In a multihop network a pair of sensors may be located outside of their respective transmission ranges and thus be unable to communicate directly. For this reason multihop networks require routing protocols [9] in which all sensors cooperate to forward messages to their destinations.

Sensors collect information about the surrounding environment (sensor field) and provide sensed data to external *sink* nodes. The sensor network can be programmed according to different paradigms which define the communication model among the sensors and between the sensor network and the sinks. In the data diffusion paradigm [8] the network is organized into a directed acyclic graph rooted at the sink, and sensed data flow with different rates from sensors to the sink. In recent paradigms [11, 12] the network is seen as a database and the sink programs the network by sending queries to sensors.

In many cases sensors rely on batteries for their operation. Consequently efficient use of energy is an important issue [15] and several techniques have been proposed that attempt to prolong battery lifetime by exploiting energy efficient communication strategies.

Sensors can periodically send data, either directly acquired or generated from acquired information, to other nodes. The communication succeeds only if the radio interfaces of both the transmitting and the receiving nodes are simultaneously active. The trivial solution to guarantee this is to maintain the radios of all nodes in idle mode (that is, ready to receive data). This strategy allows any node to receive any data when the sender arbitrarily decides to transmit it. However, maintaining the radio interface continuously active implies a relevant consumption of energy with the consequent dramatic reduction of network lifetime. One of the most important sources of energy inefficiency in wireless communication is idle mode radio power consumption. A radio in idle mode draws almost the same power as when it is in receive and transmit modes [15].

An alternative approach is to synchronize all nodes, or adjacent nodes, to activate their interfaces simultaneously at specified regular time intervals. This approach reduces energy consumption, however, in several cases many nodes switch on their interface unnecessarily, since no data is sent to them.

These techniques for synchronized radio activity typically operate at the MAC layer or exploit coordinated activities of the network and MAC layers. In any case, they do not exploit information from the application layer. When the application layer of some node wishes to send data to one of its neighbors, it has to wait until its next activation interval, introducing latency in the communication. Moreover, if the data must cross a multihop path, it could be delayed several times due to the periods of inactivity of nodes in the path.

In this paper we consider an energy efficient communication model for sensor networks which is driven by the application. It is suitable for applications where sensors communicate by sending data streams to each other (as it is the case in data diffusion), and it exploits information from the application layer to synchronize the sensors along a communication channel in order to forward the data stream to the destination. In our model the time axis is divided into periods. Each period comprises a *system window* in which the application constructs communication channels and a *communication window*. In the communication window only sensors belonging to a communication channel are active. These nodes are synchronized, exploiting application driven information, in order to forward data streams to their destination with the minimum possible latency. In system windows all the sensors are active in order to provide connectivity between an arbitrary pair of sensors. Note that MAC layer solutions can be used during system windows to further reduce power consumption.

We propose two strategies to coordinate the sensors in a certain communication channel in order to reduce the time they should have the radio on. Focusing on a communication channel, we give a probabilistic model under which the probability of packet loss can be analytically evaluated and we provide a cost model to evaluate the energy efficiency of the two strategies. Then we discuss how the strategies can be tuned according to different requirements of the application.

The rest of the paper is organized as follows. Section 2 reviews related work. In Section 3 we presents an applicative scenario. Section 4 describes our strategies. Section 5 analyzes the cost of the proposed solutions under a probabilistic model and Section 6 presents numerical results and provides a discussion. Simulation results appear in Section 7 and conclusions are drawn in Section 8.

## 2 Related Work

Among the MAC-layer solutions, [16],[17], and [14] require that each sensor be equipped with an additional radio which is reserved for signaling and which is used to synchronize communication with the main radio.

MAC solutions exploiting synchronization protocols can be found in [18], [22], and [23]. In these protocols sensors divide their time axis into frames. Each frame is divided into an *active window* (during which the radio is on) and a *power-save window* (in which the radio is off). Typically the size of the frame and the windows is a protocol-defined constant but each node selects the frame start-time independently of the other nodes, based on its local clock. The protocols differ in the way they define the windows and in the algorithms they use to synchronize the sensors.

In [18] each sensor sends a beacon within some of the active windows to communicate its clock to its neighbors. The size and structure of the active window is defined in order to let each sensor eventually receive the beacons from its neighbors. Also, in S-MAC [22],[23] sensors periodically send beacons containing their schedules (start-time of the frame). At startup, sensors try to coordinate with their neighbors in order to use the same schedule. This is not always possible, however, and some sensors have to follow multiple schedules. In order to detect new neighbors, sensors periodically keep the radio on for an entire frame. An extension to S-MAC where the size of the active window may change dynamically is described in [19].

A different approach combining information between MAC and network layer has been introduced in [24]. The scheme is not bound to any actual routing or MAC protocol but the authors assume that sensors can be in one of *active* and *power-save* modes and transitions are triggered by network-layer events and expiration of soft-state timers. Events consist in the reception of network layer messages, and result in a radio state transition depending on their meaning (for example if it is possible to infer that some messages will be received shortly).

GAF [20] and Span [3] are two approaches based on clustering. In particular they select a connected backbone of sensors which ensures network connectivity. Backbone sensors keep the radio on to route and buffer messages while the others alternate between active and power-save state. GAF [20] requires that sensors be equipped with GPS and uses location information to determine a grid-based partitioning of the network such that any two nodes in adjacent areas can communicate with each other (i.e. they are within each other's radio range). At any given time only one sensor per area is active (the area leader) and the set of area leaders constitutes the backbone. Non-active nodes periodically wake up in order to send/receive messages and elect the new area leader. Differently from GAF, Span [3] does not require GPS and uses a distributed protocol to construct the network backbone.

# 3 Scenario

There are several applications of wireless sensor networks where it is possible determine in advance the time intervals where pairs, or even groups, of nodes need to communicate. In these cases, as proposed and analyzed in this paper, it is possible to apply synchronization strategies that require nodes to activate their interfaces just when and where really needed.

As a simple example, consider the case of a node $u_1$ of wireless sensor network that has been instructed to acquire information through its transducers at a rate $r$. The information acquired is analyzed and if some condition is verified, the information is sent to node $u_2$. This, for instance, happens with the diffusion paradigm [8], if we suppose that $u_2$ is the sink node that has expressed an interest that can be satisfied by node $u_1$. If we know the instant in which $u_1$ starts to sample data we can instruct the set of nodes involved in the path from $u_1$ to $u_2$ to switch their radios on just when it is likely that they receive something. Suppose for instance, that we ask $u_1$ to sample the temperature every two seconds, and to send it to $u_2$ when it is greater that 5. In the ideal case of no communication delays, if we know the instant $t_0$ that $u_1$ starts sampling, we can ask the nodes involved in the path to switch on their radio every two seconds, starting at $t_0$, and to maintain it active for the time needed to receive the data. In the more realistic case of network delays during communication, the activation interval in the various nodes needs to be accurately adjusted, as discussed in this paper, in order to minimize the possibility of loosing packets. Note that with this schema it can still happen that some nodes activate their interface unnecessarily, for instance when the sensed temperature is below 5, however this is just limited to the nodes involved in the path from $u_1$ to $u_2$.

Similar situations, where communication rate among groups of nodes can be determined in advance, occur also in case of management of sensed data with the database paradigm [11, 21]. Consider for instance a query like "compute and send to the sink node the average temperature over a ten-minutes period, sampling temperature every thirty seconds in sensors belonging to the area A". User queries specify, among other things, information that controls the sensing rate activity and the communication rate activity. In addition, specific query processing strategies allow the precise determination of the set of nodes involved in the communication related to a query.

# 4 Communication Paradigm

As we discussed in Section 3, in several cases it is possible to know in advance when applications need to send data through the wireless network. Typically applications of wireless sensor networks request data at specific rates, from specific nodes.

This knowledge can be easily exploited in order to reduce energy consumption in single-hop communication. In this case it is sufficient that the sender and receivers be synchronized and turn on their radio interface simultaneously at, application dependent, predefined intervals for the time needed to send and receive a data packet.

However, the situation is much more difficult to handle in a multihop scenario. Given that delays may occur in the various nodes traversed by a packet, it is not so obvious how to decide when intermediate nodes should turn their radios on, to support communication between two endpoints.

## 4.1 Communication Protocol

In the paradigm we propose, when the application layer on node $u_0$ needs to send some data to node $u_n$, it requests the network layer to setup a communication channel. As detailed later, the network layer sends a *connect* message that travels towards $u_n$ and establishes a data path. Nodes in this path will configure their radio activity to support communication between $u_0$ and $u_n$. Note that only the nodes in the communication path must turn on their radio when needed, while all other nodes may keep it off.

Given that connect messages are obviously sent before the channel is set up, there must be some time periods when connect and other system messages are guaranteed to be received. To this end we suppose the existence of *system windows* when all nodes turn on their radio and wait to receive and service any connect requests. We impose a large time interval between consecutive system windows, to avoid significantly affecting network energy efficiency. Of course, this introduces some latency in the channel setup. However, in the applications we envision, connect requests are much less frequent than packet transmissions through an open channel and we are willing to accept some latency at channel setup if we substantially reduce latency and energy consumption during actual communication. Note that system windows can also be used to reconstruct channels which got broken due to node failures. Clock synchronization allows sensor nodes to have a uniform knowledge of the beginning of system windows (it is also needed for protocol operation, see below). Synchronization protocols for sensor networks can be found in [4, 13]. In the rest of the paper we disregard the issues related to clock synchronization and assume that all nodes turn on their radios at the beginning of system windows to send/forward/receive system messages.

When the application requests to set up a channel, it specifies the destination node ($u_n$), together with the size $s$ of the packets that will be sent, the time interval $t_d$ between successive transmissions of packets along the channel, and the time $t_0$, when the first packet will be sent. The network layer sends a connect message including tuple $(id, u_0, u_n, s, t_d, t_0)$ (where $id$ is a unique channel identifier). A multihop routing protocol determines intermediate nodes $u_1, u_2, \ldots, u_{n-1}$. $u_n$ acknowledges the successful reception of the connect message with a *connect-ack* which travels back to the source through the same forward path. Intermediate nodes associate the channel id with the identifiers of the previous and next hops in the channel. They will use the channel id to determine the next hop and forward incoming messages. The previous hop may be used for control information flowing back towards the source.

A problem that every node involved in a channel should solve, during channel setup, is the determination of its radio activity to reliably forward messages to the next node, i.e. every node should decide when and how long it should turn its radio on. A naive solution to this problem is as follows.

The transmission duration, $t_m$, of a packet can be estimated from the size $s$ of the packets and the radio data rate. Assume $u_0$ is supposed to start transmitting a packet at time $t_0$. If we disregard delays, $u_0$ actually starts transmitting at $t_0$ and $u_1$ starts receiving at the same time. The transmission terminates at $t_0 + t_m$ and $u_1$ immediately starts the forwarding which terminates at $t_0 + 2t_m$. Reasoning like this, we see that $u_0$ operates its radio in transmission mode over interval $[t_0, t_0 + t_m]$, any intermediate node $u_i$ operates its radio in receive mode over $[t_0 + (i-1)t_m, t_0 + i \cdot t_m]$ and in transmission mode over $[t_0 + i \cdot t_m, t_0 + (i+1)t_m]$, and $u_n$, that only receives the packet, has its radio on in receive mode over $[t_0 + (n-1)t_m, t_0 + nt_m]$. Figure 1(a) illustrates this simple scenario for a path of length 4. All nodes will repeat this process every $t_d$.

The situation we presented is totally unrealistic since medium contention delays cannot be disregarded. Node $u_0$ won't be able to immediately acquire the medium at time $t_0$. Consequently, the actual transmission begins some time later. Correspondingly, $u_1$ starts receiving later. Also, when $u_1$ attempts to retransmit the packet, it will incur some delays. Delays propagate down the path and result in delayed transmissions and receptions of packets (see Figure 1(b)).

The radio activity intervals of the nodes should be adjusted taking delays into account. We will use the following strategy:

1. A node switches its radio on (idle mode) at the earliest possible time (i.e. to be ready to receive the packet in case there are no delays). As derived above, the switch on times for $u_i$ ($i \in \{1, 2, \ldots, n\}$) are $t_0 + k \cdot t_d + (i-1)t_m$.

2. When a packet arrives, $u_i$ receives it (receive mode), forwards the packet to the next node (transmit mode) or passes it to the application (if $i = n$) and turns off the radio.

3. A node $u_i$ ($i \in \{1, 2, \ldots, n\}$) will switch the radio off, in case no packet has arrived, at times $t_0 + k \cdot t_d + (i-1)t_m + t_i^*$ (after which it supposes that the packet got lost).

The critical point is how to determine the $t_i^*$s, in an optimal way: the radio should be kept in idle mode long enough to guarantee, with a certain probability, that packets reach the destination and, at the same time, save as much energy as possible. We analyze two strategies. In the first case $t_i^*$ is the same for all nodes in the path. In the second case it increases proportionally to $i$. More formally:

***Fix:*** $t_i^* = t^*$, for some $t^* \in [0, t_d]$

***Lin:*** $t_i^* = i \cdot t^{**}$, for some $t^{**} \in [0, \frac{t_d}{n}]$

The next Section will model these two strategies using a probabilistic framework which will permit the determination of $t^*$ and $t^{**}$ that ensure a certain packet delivery probability. The probabilistic framework will also be used to derive a cost model that estimates the energy consumption of the two strategies, depending on the reliability of the communication channel.

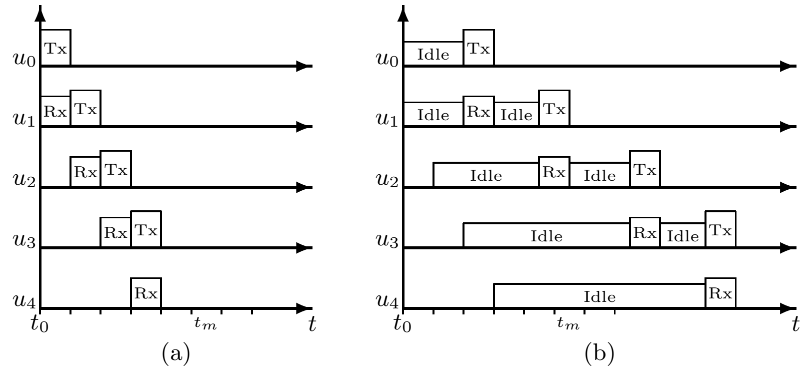



Figure 1: Radio activity for nodes in a path: (a) simple, naive model; (b) delays-aware model. Tx = transmission times, Rx = reception times, Idle = idle times due to accumulated delays.

# 5 Analysis

## 5.1 Probabilistic Model

Every node may incur some delays before being able to actually send a packet. Internal processing and, chiefly, contention for access to the shared medium are the causes of these delays. We model the total delay introduced by node $u_i, i \in \{0, 1, \ldots, n-1\}$, with a continuous random variable $D_i$. In order to simplify the analysis, we assume that the $D_i$s are identically distributed independent random variables. We also define

$$R_m \quad = \quad \sum_{i=0}^{m-1} D_i$$

the total delay accumulated by nodes $u_0, u_1, \ldots, u_{m-1}$, for $m \in \{1, 2, \ldots, n\}$.

In addition, we model the possibility of losing messages due to various problems such as resource shortages at a certain node (full queues, etc.) and radio interferences. To this end, we define random variables $X_0, X_1, \ldots, X_n$ identically distributed according to the following rule:

$$\begin{aligned} \mathcal{P}(X_i = 1) &= p \\ \mathcal{P}(X_i = 0) &= 1 - p \end{aligned}$$

where $p \in [0, 1]$, and say that node $u_i$ loses the packet (or does not receive it) if $X_i = 0$. We suppose that the $X_i$s are independent and are mutually independent with the $D_i$s.

In the following we will suppose that the $D_i$s have exponential densities, however the probabilistic model that we develop is general and can be instantiated with other distributions.

Given the packet loss probability $p$ and the path length $n$, we denote the probability of successfully delivering a packet to the destination using strategy *Fix* as $\mathcal{P}_S^{(F)}(p, n, t^*)$. Correspondingly we denote the probability of successfully delivering a packet to the destination using strategy *Lin* as $\mathcal{P}_S^{(L)}(p, n, t^{**})$.

Using $\mathcal{P}_S^{(F)}$ and $\mathcal{P}_S^{(L)}$ it will be possible to decide $t^*$ and $t^{**}$ such that the probability of successfully delivering a packet to destination is above a chosen threshold $\gamma$. More formally, it will possible to set $t^*$ and $t^{**}$ so that

$\mathcal{P}_S^{(F)}(p,n,t^*) > \gamma$ or $\mathcal{P}_S^{(L)}(p,n,t^{**}) > \gamma$, for some $\gamma$.

In the following we will show how to express these probabilities.

### 5.1.1 Success Probability of *Fix*

*Fix* successfully delivers the packet to the destination when all $X_i$s are 1 and $R_i < t^*$ for $i \in \{1, 2, \ldots, n\}$:

$$\begin{aligned}\mathcal{P}_S^{(F)}(p,n,t^*) &= \mathcal{P}(X_0{=}1,\ldots,X_n{=}1,R_1{\le}t^*,\ldots,R_n{\le}t^*) \\ &= \mathcal{P}(X_0{=}1,\ldots,X_n{=}1)\,\mathcal{P}(R_1{\le}t^*,\ldots,R_n{\le}t^*) \\ &= p^{n+1}\ \mathcal{P}(R_n{\le}t^*)\end{aligned} \tag{1}$$

The last passage is motivated by the fact that event $(R_n = \sum_{i=0}^{n-1} D_i \le t^*)$ implies each of $(R_1 = D_0 \le t^*)$, $(R_2 = D_0 + D_1 \le t^*)$, $\ldots$, $(R_n = \sum_{i=0}^{n-1} D_i \le t^*)$. As a consequence, $\mathcal{P}(R_1 \le t^*, \ldots, R_n \le t^*) = \mathcal{P}(R_n \le t^*)$.

If the $D_i$s are exponentially distributed with parameter $\lambda$, it is easy to show that

$$\mathcal{P}_S^{(F)}(p,n,t^*) = p^{n+1}(1{-}\exp(-\lambda t^*)\textstyle\sum_{i=0}^{n-1}\frac{(\lambda t^*)^i}{i!}) \tag{2}$$

### 5.1.2 Success Probability of *Lin*

*Lin* successfully delivers the packet to the destination when all $X_i$s are 1 and $R_i < i \cdot t^{**}$ for $i \in \{1, 2, \ldots, n\}$. Hence, we have

$$\begin{aligned}\mathcal{P}_S^{(L)}(p,n,t^{**}) &= \mathcal{P}(X_0{=}1,\ldots,X_n{=}1,R_1{\le}1\,t^{**},\ldots, \\ &\qquad R_n{\le}n\,t^{**}) \\ &= \mathcal{P}(X_0{=}1,\ldots,X_n{=}1) \\ &\qquad \mathcal{P}(R_1{\le}1\,t^{**},\ldots,R_n{\le}n\,t^{**}) \\ &= p^{n+1}\ \mathcal{P}(R_1{\le}1\,t^{**},\ldots,R_n{\le}n\,t^{**})\end{aligned} \tag{3}$$

Calculating the above probability requires the joint density function of $R_1$, $R_2, R_3, \ldots, R_n$ which might not be easy to determine. However, if we recall that $R_m = \sum_{i=0}^{m-1} D_i$, we see that event $(R_1 \le 1\,t^{**}, \ldots, R_n \le n\,t^{**})$ can be expressed in terms of $D_0, \ldots, D_{n-1}$ as $(D_0 \le 1\,t^{**}, \ldots, \sum_{i=0}^{n-1} D_i \le n\,t^{**})$. Since the $D_i$s are independent, their joint density function is $\mathbf{f}(\mathbf{x}) = f_{D_0,\ldots,D_{n-1}}(x_0, \ldots, x_{n-1}) = \prod_{i=0}^{n-1} f_{D_i}(x_i)$. We have:

$$\begin{aligned}\mathcal{P}(R_1{\le}1\,t^{**},\ldots,R_n{\le}n\,t^{**}) &= \\ \mathcal{P}(D_0{\le}1\,t^{**},\ldots,\textstyle\sum_{i=0}^{n-1} D_i{\le}n\,t^{**}) &= \int_A \mathbf{f}(\mathbf{x})\,\mathbf{dx}\end{aligned} \tag{4}$$

where $A \subseteq \mathbf{R}^n$ is the set of n-dimensional points $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ satisfying the following relations

$$\begin{cases} 0 \le x_1 \le 1\,t^{**} \\ \quad\ldots \\ 0 \le x_m \le m\,t^{**} - \sum_{i=1}^{m-1} x_i \\ \quad\ldots \\ 0 \le x_n \le n\,t^{**} - \sum_{i=1}^{n-1} x_i \end{cases}$$

Let us now assume that the $D_i$s are exponentially distributed and define $Q_m$ as the probability that a packet successfully reaches node $m$ in the channel in case packets can only get lost due to accumulated delays (i.e. $p = 1$). Formally:

$$Q_m = \mathcal{P}(D_0{\le}1\,t^{**},\ldots,\textstyle\sum_{i=0}^{m-1} D_i{\le}m\,t^{**}) \tag{5}$$

We see that $\mathbf{f}(\mathbf{x}) = \lambda^n \prod_{i=1}^{n} \exp(-\lambda x_i)$ and some algebra gives us

$$\begin{aligned} Q_1 &= 1{-}\exp(-\lambda\,t^{**}) \\ Q_m &= Q_{m-1}{-}(\lambda\,t^{**})^{m-1}\exp(-m\,\lambda\,t^{**})\left(\frac{m^{m-2}}{(m-1)!}\right) \\ &\qquad \text{for } 2{\le}m{\le}n\end{aligned} \tag{6}$$

so

$$\mathcal{P}_S^{(L)}(p,n,t^{**}) = p^{n+1} Q_n \tag{7}$$

### 5.2 Cost Estimation

We now evaluate the average cost, in terms of energy consumption, required to send a packet using strategies *Fix* and *Lin*. For comparison purposes we also evaluate the cost for strategy *Naive* that maintains the radio always in idle state (always waiting for a message).

Let $\mathcal{P}_S$ be the probability of successfully delivering a packet to the destination and let $\mathcal{C}_S$ be the relative cost. In addition, let $\mathcal{P}_{F,m}$ be the probability that the packet gets lost at node $u_m$ for some $m \in \{0, 1, \dots, n\}$ and, correspondingly, let $\mathcal{C}_{F,m}$ be the cost in that case. The average cost of sending a packet is given by

$$\mathcal{C} \quad = \quad \mathcal{P}_S\ \mathcal{C}_S + \sum_{m=0}^{n} \mathcal{P}_{F,m}\ \mathcal{C}_{F,m} \tag{8}$$

The specific probabilities and costs depend on the actual strategy used to manage the radio. In the next sub-sections we will evaluate the terms of Equation 8 for *Naive*, *Fix* and *Lin*. We will use the following notation:

$t_m$: time to send/receive a message;

$P_t$: tx mode radio power consumption;

$P_r$: rx mode radio power consumption;

$P_i$: idle mode radio power consumption;

$E_m$: total energy spent to receive and forward a message ($P_t \cdot t_m + P_r \cdot t_m$);

#### 5.2.1 Average Cost of *Naive*

*Naive* successfully delivers a packet to the destination with probability

$$\mathcal{P}_S^{(N)}(p) \quad = \quad p^{n+1} \tag{9}$$

while the probability of failing at node $u_m$ ($0 \leq m \leq n$) is

$$\mathcal{P}_{F,m}^{(N)}(p) \quad = \quad p^m(1-p) \tag{10}$$

The total energy *Naive* expends when successfully delivering a packet to the destination is

$$\mathcal{C}_S^{(N)} \quad = \quad nE_m + (n-1)P_i(t_d - 2\,t_m) + 2P_i(t_d - t_m) \tag{11}$$

where we summed the energies expended on each link in the path. In case of failure the expended energy depends on where it occurred (i.e. the first node failing to receive the packet). Upstream nodes transmit and receive the packet while downstream nodes do not. We have:

$$\begin{aligned} \mathcal{C}_{F,0}^{(N)} \quad &= \quad (n+1)P_i\,t_d \qquad (12)\\ \mathcal{C}_{F,m}^{(N)} \quad &= \quad mE_m + 2P_i(t_d - t_m) + (m-1)P_i(t_d - 2\,t_m) + \\ &\qquad (n-m)P_i\,t_d \\ &\qquad\qquad \text{for } 1 \leq m \leq n \end{aligned}$$

For simplicity, we assumed that the radio operates in idle mode when contending for medium access, even if this might not be the case in reality.

The estimation of the cost attributable to *Naive* for the transmission of one packet can now be evaluated according to Equation 8.

#### 5.2.2 Average Cost of *Fix*

*Fix* successfully delivers a packet to node $u_n$ with probability (Equation 1):

$$\mathcal{P}_S^{(F)}(p) \quad = \quad p^{n+1}\ \mathcal{P}(R_n \leq t^*) \tag{13}$$

For the probabilities of failure we have:

$$\begin{aligned} \mathcal{P}_{F,0}^{(F)}(p) \quad &= \quad \mathcal{P}(X_0=0) \ = \ 1-p \qquad (14)\\ \mathcal{P}_{F,1}^{(F)}(p) \quad &= \quad \mathcal{P}(X_0=1, (X_1=0 \vee (X_1=1, R_1 > t_1^*)) \end{aligned}$$

$$
\begin{aligned}
&= p - p^2\ \mathcal{P}(R_1 \leq t_1^*) \\
\mathcal{P}_{F,m}^{(F)}(p) &= \mathcal{P}(X_0{=}1, X_1{=}1, \ldots, X_{m-1}{=}1, R_{m-1} \leq t^*, \\
&\qquad (X_m{=}0 \vee X_m{=}1, R_m > t^*)) \\
&= p^m\ \mathcal{P}(R_{m-1} \leq t^*) - p^{m+1}\ \mathcal{P}(R_m \leq t^*) \\
&\qquad \text{for } 2 \leq m \leq n
\end{aligned}
$$

Successfully delivering a packet to node $u_n$ costs

$$
\mathcal{C}_S^{(F)} = nE_m + P_i\ \mathcal{E}(D_0 | R_n \leq t^*) \left( \frac{n(n+3)}{2} \right) \tag{15}
$$

For the costs of failure we give the following upper bounds

$$
\begin{aligned}
\mathcal{C}_{F,0}^{(F)} &\leq nP_i\, t^* \\
\mathcal{C}_{F,1}^{(F)} &\leq E_m + (n+1)P_i\, t^* \\
\mathcal{C}_{F,m}^{(F)} &\leq mE_m + (n-m+2)P_i\, t^* + \frac{m(m-1)}{2} P_i\ \mathcal{E}(D_j | R_{m-1} \leq t^*) \\
&\qquad \text{for } 2 \leq m \leq n
\end{aligned} \tag{16}
$$

In case the $D_i$s have exponential densities with parameter $\lambda$ and $h \geq 1, j \in \{0, \ldots, h-1\}, \tau > 0$, we have

$$
\mathcal{P}(R_h \leq \tau) = 1 - \exp(-\lambda\tau) \sum_{i=0}^{h-1} \frac{\lambda^i \tau^i}{i!}
$$

and

$$
\mathcal{E}(D_j | R_h \leq \tau) = \frac{1}{\lambda} \left( \frac{1 - exp(-\lambda\tau) \sum_{i=0}^{h} \frac{(\lambda\tau)^i}{i!}}{1 - exp(-\lambda\tau) \sum_{i=0}^{h-1} \frac{(\lambda\tau)^i}{i!}} \right) \tag{17}
$$

This allows the evaluation of Equations 13 - 16. The estimation of the cost attributable to *Fix* for the transmission of one packet can now be derived introducing Equations 13-16 in Equation 8.

#### 5.2.3 Average Cost of *Lin*

For *Lin* we define $A_m$ as the event that a packet successfully reaches node $m$ in the channel in case packets can only get lost due to accumulated delays (i.e. $p = 1$):

$$
A_m = (R_1 \leq t^{**}, R_2 \leq 2\, t^{**}, \ldots, R_m \leq m\, t^{**})
$$

for $m \in \{1, 2, \ldots, n\}$. Recalling Equation 3, we may write

$$
\mathcal{P}_S^{(L)}(p, n, t^{**}) = p^{n+1}\, \mathcal{P}(A_n) \tag{18}
$$

For the probabilities of failure we have:

$$
\begin{aligned}
\mathcal{P}_{F,0}^{(L)}(p) &= \mathcal{P}(X_0{=}0) = 1 - p \\
\mathcal{P}_{F,1}^{(L)}(p) &= \mathcal{P}(X_0{=}1, (X_1{=}0 \vee (X_1{=}1, R_1 > t^{**})) \\
&= p - p^2\ \mathcal{P}(A_1) \\
\mathcal{P}_{F,m}^{(L)}(p) &= \mathcal{P}(X_0{=}1, X_1{=}1, \ldots, X_{m-1}{=}1, A_{m-1}, \\
&\qquad (X_m{=}0 \vee (X_m{=}1, R_m > m\, t^{**})) \\
&= p^m\ \mathcal{P}(A_{m-1}) - p^{m+1}\ \mathcal{P}(A_m) \\
&\qquad \text{for } 2 \leq m \leq n
\end{aligned} \tag{19}
$$

Similarly to *Fix*, the cost for a successful transmission to node $u_n$ is

$$
\mathcal{C}_S^{(L)} = nE_m + P_i\ \mathcal{E}(D_0 | R_n \leq n\, t^{**}) \left( \frac{n(n+3)}{2} \right) \tag{20}
$$

For the costs of failure we give the following upper bounds

$$
\begin{aligned}
\mathcal{C}_{F,0}^{(L)} &\leq (\textstyle\sum_{i=0}^{n} i) P_i\, t^{**} \\
\mathcal{C}_{F,1}^{(L)} &\leq E_m + (\textstyle\sum_{i=0}^{n} i + 1) P_i\, t^{**} \\
\mathcal{C}_{F,m}^{(L)} &\leq mE_m + (\textstyle\sum_{i=m}^{n} i + m) P_i\, t^{**} + \\
&\qquad (\textstyle\sum_{i=0}^{m-1} i) P_i\ \mathcal{E}(D_0 | R_{m-1} \leq (m-1)\, t^{**}) \\
&\qquad \text{for } 2 \leq m \leq n
\end{aligned} \tag{21}
$$

When the $D_i$s have exponential densities with parameter $\lambda$, we observe that $\mathcal{P}(A_m) = Q_m$ (Equations 5 - 6), so introducing Equation 6 in Equations 18,19 yields the probabilities for *Lin*. Introducing Equation 17 in Equations 20,21 yields the related costs. Again, the estimation of the cost attributable to *Lin* for the transmission of one packet can finally be derived introducing Equations 18 - 21 in Equation 8.

| | $n$ | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\lambda$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 15 |
| $\frac{1}{3.125}$ | 17 | 24 | 29 | 35 | 40 | 45 | 49 | 54 | 59 | 63 | 84 |
| $\frac{1}{6.25}$ | 34 | 47 | 58 | 69 | 79 | 89 | 98 | 108 | 117 | 125 | 168 |
| $\frac{1}{12.5}$ | 67 | 93 | 116 | 138 | 158 | 177 | 196 | 215 | 233 | 250 | 336 |
| $\frac{1}{25}$ | 133 | 186 | 232 | 275 | 315 | 354 | 392 | 429 | 465 | 500 | 671 |
| $\frac{1}{50}$ | 265 | 372 | 464 | 549 | 630 | 708 | 783 | 857 | 929 | 1000 | 1342 |
| $\frac{1}{100}$ | 530 | 744 | 928 | 1098 | 1260 | 1415 | 1566 | 1714 | 1858 | 2000 | 2684 |

Table 1: Optimum values for parameter $t^*$ for success delivery probability of 0.995 ($p = 1$)

# 6 Numerical Results

The analysis from the previous Section suggests a tradeoff between the probability of successfully delivering a packet along a channel and the the cost, in terms of energy consumption, of strategies *Fix* and *Lin*. The length of the radio activity interval for each of the nodes in the channel affects this tradeoff. In particular, as the interval length increases, both the cost and the probability of successful delivery increase, and, as the interval length decreases, both the cost and the success probability decrease. Note that, since the probability of packet loss is directly related to the quality of service (QoS) given to the application, this tradeoff is between QoS and cost.

In order to numerically evaluate our strategies, we propose a solution to this tradeoff. We select a threshold $\gamma$ on the success packet delivery probability and identify the smallest radio activity interval of each node that assures that the probability of successful delivery exceeds $\gamma$. In particular, we assume an exponential density function for the random variables $D_i$s representing delays on the path and determine parameters $t^*$, for *Fix*, and $t^{**}$, for *Lin*, as a function of $\gamma$ and $n$ (the channel length) from equations 2 and 7.

Next, we plot the energy savings that our strategies can achieve with respect to *Naive* and discuss the results.

## 6.1 Configuration

We begin with the calculation of the optimal values of parameters $t^*$ and $t^{**}$ for *Fix* and *Lin*. We plotted $\mathcal{P}_S^{(F)}(1, n, t^*)$ from Equation 2 for different path lengths, $n$, and $\lambda$ values and found the minimum value of $t^*$ that rendered the above probability greater than $\gamma = 0.995$.

In other words, our aim was to locate, for several values of $n$ and $\lambda$, the minimum $t^*$ for which *Fix* can successfully deliver a packet to the destination with high enough probability. We only considered delays introduced by the nodes as the possible causes of failure and assumed that nodes never lost a packet for other reasons (i.e. $p = 1$). Table 1 presents the results we obtained and Figure 2 represents the data graphically.

The Table clearly indicates that, for a fixed path length, halving the value of $\lambda$ doubles the optimum $t^*$. On the other hand, for a fixed $\lambda$, $t^*$ does not grow linearly with $n$ but a linear approximation can be acceptable.

We plotted $\mathcal{P}_S^{(L)}(1, n, t^{**})$ from Equations 18 and 6 for the same $n$ and $\lambda$ values we used for *Fix* and found the minimum value of $t^{**}$ that rendered the probability greater than $\gamma = 0.995$. The results are shown in Table 2.

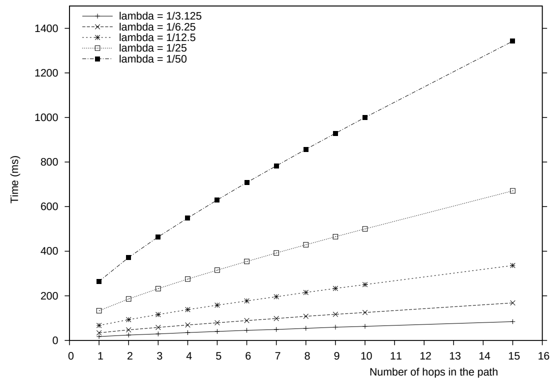



Figure 2: Optimum values for parameter $t^*$ for success delivery probability of 0.995 ($p = 1$)

As can be seen, $t^{**}$ doesn't change with increasing path lengths. This might be counter-intuitive but, when we looked at the graphs of $\mathcal{P}_S^{(L)}(1, n, t^{**})$ for a fixed $\lambda$ and growing values of $n$, we observed that the central part of the curves moves to the right (the minimum value of $t^{**}$ to have a 0.5 probability of delivering the packet to the destination grows with $n$). However, there is no noticeable change in $t^{**}$ when we aim at a 0.995 probability. As for *Fix*, halving $\lambda$ for a fixed $n$ doubles $t^{**}$.

| | $n$ | |
|---|---|---|
| $\lambda$ | 1 | 2-15 |
| $\frac{1}{3.125}$ | 17 | 17 |
| $\frac{1}{6.25}$ | 34 | 34 |
| $\frac{1}{12.5}$ | 67 | 67 |
| $\frac{1}{25}$ | 133 | 134 |
| $\frac{1}{50}$ | 265 | 267 |
| $\frac{1}{100}$ | 530 | 533 |

Table 2: Optimum values for parameter $t^{**}$ for success delivery probability of 0.995 ($p = 1$)

## 6.2 Costs

The next step in our numerical evaluation is to compare the algorithms according to the analytical formulas we developed for the costs (Section 5.2). We assume sensors equipped with a 12.4 kb/s radio with $P_t = 36.0$ mW, $P_r = 30.0$ mW, $P_i = 24.0$ mW and consider packets of 40 bytes (320 bits), resulting in $t_m = 25.806$ ms, $E_m = 1703.226$ $\mu$J. Note that these settings are typical of low-end sensors like the mica2 motes [2].

Figure 3 presents several graphs for $\frac{\mathcal{C}^{(F)}(p)}{\mathcal{C}^{(N)}(p)}$ and $\frac{\mathcal{C}^{(L)}(p)}{\mathcal{C}^{(N)}(p)}$ for a path of length 6 and different values of $\lambda$ and $t_d$. Figure 3(a) plots the above ratios for $\lambda = \frac{1}{6.25}$ and $t_d = 2000$ ms while Figure 3(b) gives the results for the same value of $\lambda$ and $t_d = 4000$ ms. Figures 3(c)-(d) and (e)-(f) propose the same couple of $t_d$ values for $\lambda = \frac{1}{12.5}$ and $\lambda = \frac{1}{25}$, respectively.

Figures 3(a)-(b) show that, when medium contention delays are very small the two algorithms compare almost equal and achieve savings of about 95-98% with respect to *Naive*. The curves are almost flat, which indicates that message communication dominates medium contention delays. Figures 3(c)-(f) give clear evidence that *Fix* achieves greater savings than *Lin*, when $p < 1$ (the algorithms compare equal when $p \approx 1$).

If we (very roughly) approximate the curve for $\lambda = \frac{1}{25}$ in Figure 2 with $y = 38x + 118$ and imagine a path of $n$ hops, we calculate the cost for a failure at node $u_0$, for *Fix*, as

$$c_1 = n(38n + 118) = 38n^2 + 118n$$

while, for *Lin*, we have

$$c_2 = 134(1 + 2 + \cdots + n) = 67n^2 + 67n$$

We easily find out that $c_1 < c_2$ iff $n \geq 2$ while Tables 1 and 2 tell us that $c_1 = c_2$ if $n = 1$. If the failure occurs at node $\overline{n}$ ($\overline{n} > 0$), a slightly more laborious calculation shows that, after that node, *Lin* always costs more than *Fix*. Assuming comparable costs before the point of failure, we see that *Lin* always costs more than *Fix*.

When $p \approx 1$ the probability of losing a packet along the way is very small since extremely large delays are infrequent, and the two algorithms have roughly the same cost. As $p$ decreases, however, more packets get lost and, as the previous arguments suggest, nodes that do not receive a packet consume more energy in *Lin* than they do in *Fix*.

Figure 3(a) is very similar to Figure 3(d). This is reasonable since Tables 1 and 2 indicate that $t^*$ and $t^{**}$ double when $\lambda$ halves. Since in going from (a) to (b), $t_d$ also doubles, the cost ratios stay the same. The same phenomenon explains the similarity between Figures 3(c) and (f).

Figure 3 as a whole also illustrates the general trend that the relative cost efficiency of both *Fix* and *Lin* decreases as the average per-node delay increases. Energy savings decrease more rapidly if $p < 1$.

## 7 Simulations

We implemented our strategies on the nesC/TinyOs platform [5] [7] and ran simulations with Tossim [10], the simulator for mica motes [6]. Our goal was to verify whether the assumptions we made in the probabilistic model are actually applicable in a sensor network. We had to modify (in straightforward ways) some of the TinyOs modules in order to obtain millisecond accuracy in the measurements and render the MAC scheme compatible with the strategies we propose.

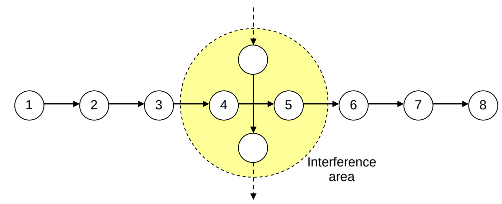


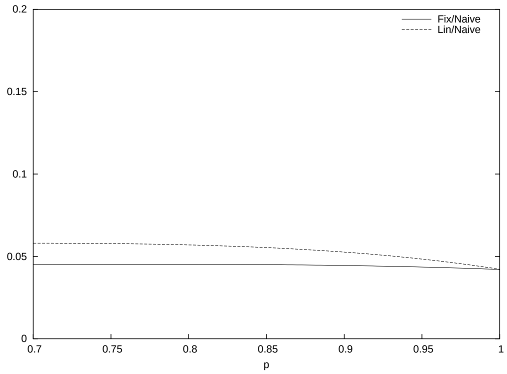

Figure 4: Simulation topology sample

We set up a channel of 8 nodes and simulated a fixed data rate using packets with a 20-byte payload. We measured the number of packets lost on the path and, for each node, the medium contention delay introduced and the radio activity interval for several values of the parameters $t^*$ (for *Fix*) and $t^{**}$ (for *Lin*).

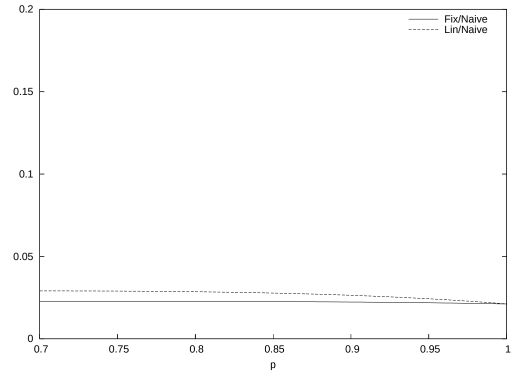
In order to simulate interferences and packet losses we deployed a second channel and let it interfere with the first. In each of several scenarios we simulated, the second channel interferes with the first on one of its links. Figure 4 illustrates the scenario where the interference occurs on the link between nodes 4 and 5 of the primary (measured) channel. On each channel each node is in radio range of only the previous and next hop nodes. Nodes in the region of interference (the circled area in Figure 4) can hear each other. We also ran a simulation where the primary channel suffers no interference.

The primary channel has a data rate of 1 packet every second (a total of 300 packets) versus 1 packet every 0.8 seconds for the interfering channel (375 packets).

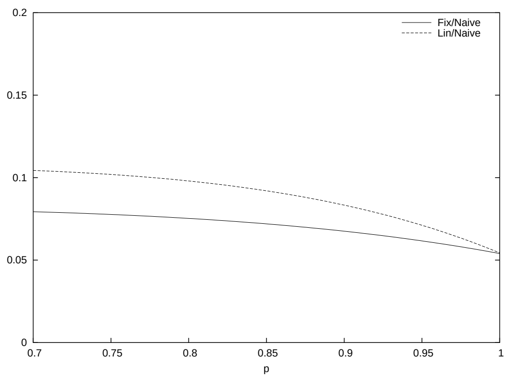
Given the results from all the simulated scenarios, we computed a weighted average of the data. The non-interference scenario had a weight of $\frac{1}{2}$ while each of the other scenarios had a weight of $\frac{1}{14}$.



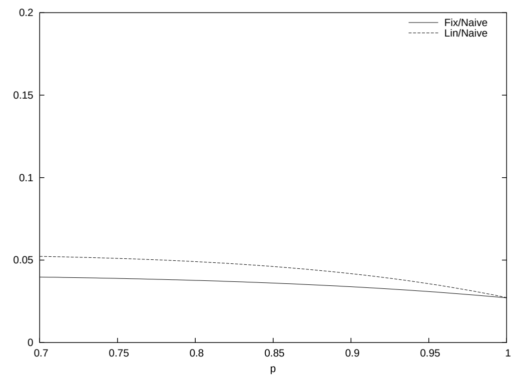

(a) $\lambda = \frac{1}{6.25}, t_d = 2000$



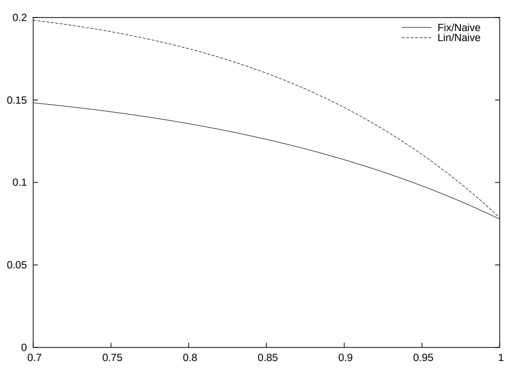

(b) $\lambda = \frac{1}{6.25}, t_d = 4000$



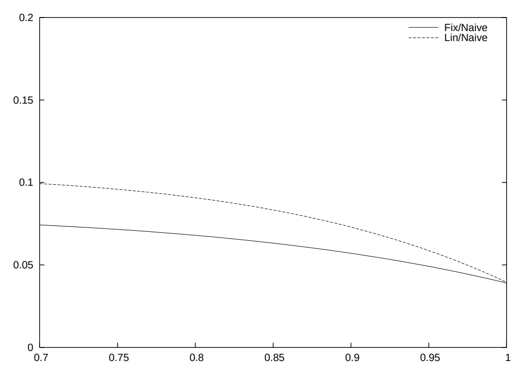

(c) $\lambda = \frac{1}{12.5}, t_d = 2000$




(d) $\lambda = \frac{1}{12.5}, t_d = 4000$

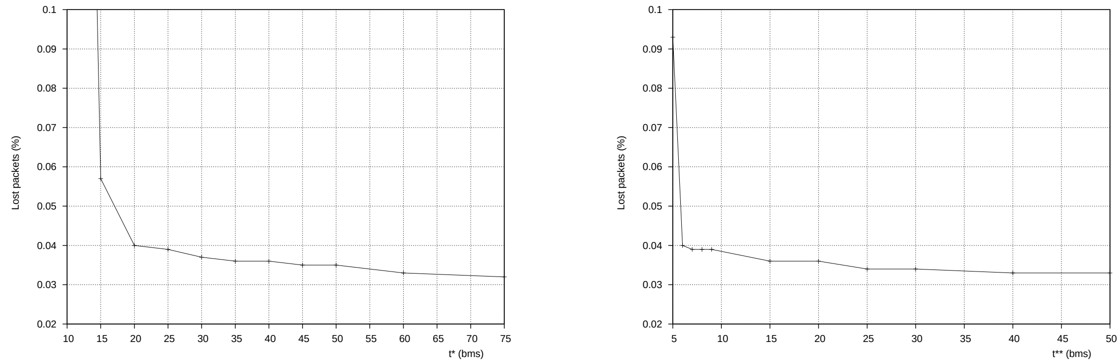



(e) $\lambda = \frac{1}{25}, t_d = 2000$



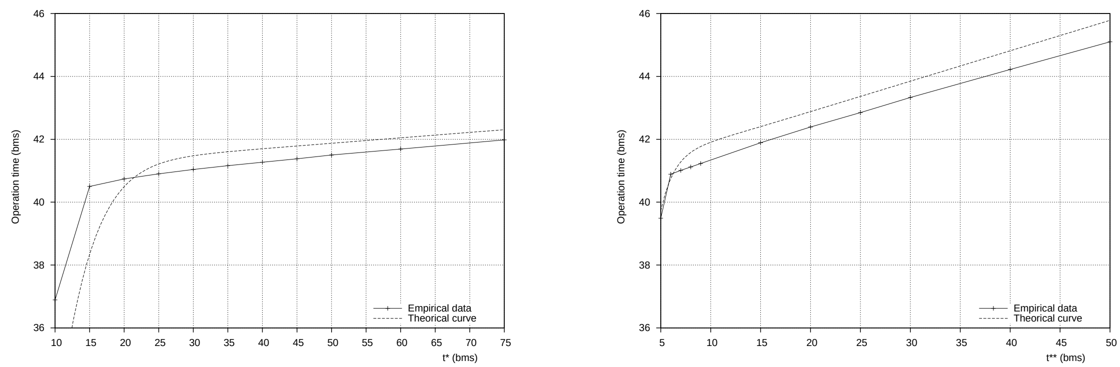

(f) $\lambda = \frac{1}{25}, t_d = 4000$

Figure 3: Comparison of cost functions for *Naive*, *Fix* and *Lin* for different path settings.




(a) *Fix* (b) *Lin*

Figure 5: Lost packets percentage.




(a) *Fix* (b) *Lin*

Figure 6: Radio operation times.

Figure 5 reports the average percentage of lost packets for *Fix* (left) and *Lin* (right) for several values of the parameters $t^*$ and $t^{**}$ in binary milliseconds (bms). The graphs indicate that there is a minimum percentage (0.032) of lost packets that cannot be avoided. An accurate analysis of the simulation traces revealed that these packets got lost due to the hidden terminal problem. The mica radio stack (which Tossim simulates) has no RTS/CTS mechanism to tackle this problem.

The last graphs we present (Figure 6) compare the average radio activity obtained from the simulations with the average radio activity of a node derived from the analytical model with $t^* \in [10, 75]$ for *Fix* (left) and $t^{**} \in [5, 50]$ for *Lin* (right).

The analytical curves were derived assuming an exponential density for the $D_i$s. Since $\mathcal{E}(D_i) = \frac{1}{\lambda}$, we evaluated the expected value $\mu$ of the empirical density and derived the value of $\lambda$ for the exponential density as $\lambda = \frac{1}{\mu}$. The value of parameter $p$ was calculated from the physiological loss probability that we measured in the simulations.

Comparing the empirical data from Figure 6, we observe that for values of $t^*$ and $t^{**}$ that ensure the same low loss probability (say $t^* = 60$ bms and $t^{**} = 40$ bms, from Figure 5), nodes operate the radio longer in *Lin* than they do in *Fix*. This is in accordance with the results we found in the analysis.

# 8 Conclusions

We have considered an application-driven communication model for sensor networks. The model exploits information from the application layer in order to set up communication channels between pairs of sensors, along which a data stream travels with rates defined by the application. Focusing on a single communication channel, we have introduced two strategies which attempt to synchronize the sensors in the channel in order to save energy. We have proposed a probabilistic model under which the probability of packet loss can be analytically evaluated and we have given a cost model to evaluate the energy efficiency of the two strategies. Based on the cost and packet loss probabilistic model, we have shown how the sensors in the channel can be configured in order to achieve a desired threshold of successful packet delivery probability and to minimize the cost. Lastly, we have validated and confirmed the accuracy of our model with a simulation made with Tossim (the Berkeley motes' simulator). Future work includes comparing the latency of our approach with those achieved by energy-efficient MAC/network layer protocols for sensor networks. We also plan to evaluate the cost of the considered model on a network in which several channels can be active at the same time. Another research direction is related to the determination of the probability distributions which better model the delay introduced by each node in a given channel.

# References


[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci. "Wireless Sensor Networks: A Survey". *Computer Networks*, 38(4), March 2002, pp. 393-422.

[2] Chipcon. CC1000 RF Transceiver Data Sheet. http://www.chipcon.com/files/CC1000_Data_Sheet_2_2.pdf

[3] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, Robert Morris. "Span: An Energy Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks". *ACM Wireless Networks Journal*, 8(5), September 2002, pp. 481-494.

[4] Saurabh Ganeriwal, Ram Kumar, Mani B. Srivastava. "Timing-sync Protocol for Sensor Networks". *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys 2003)*, Los Angeles, CA, USA, November 2003, pp. 138-149.

[5] David Gay, Philip Levis, Robert von Behren, Matt Welsh, Eric Brewer, David Culler. "The nesC Language: A Holistic Approach to Networked Embedded Systems". *Proceedings of the 2003 0ACM SIGPLAN Conference on Programming Lan-*

*guage Design and Implementation (PLDI 2003)*, San Diego, CA, USA, June 2003, pp 1-11.

[6] Jason L. Hill, David E. Culler. "MICA: A Wireless Platform for Deeply Embedded Networks". *IEEE Micro*, 22(6), Nov/Dec 2002, pp. 12-24.

[7] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, Kristofer Pister. "System Architecture Directions for Networked Sensors". *ACM SIGARCH Computer Architecture News*, 28(5), December 2000, pp. 93-104.

[8] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin. "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks". *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, Boston, MA, USA, August 2000, pp. 56-67.

[9] Brad Karp, H. T. Kung. "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks". *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, Boston, MA, USA, August 2000, pp. 243-254.

[10] Philip Levis, Nelson Lee, Matt Welsh, David Culler. "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications". *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys 2003)*, Los Angeles, CA, USA, November 2003, pp. 126-137.

[11] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, Wei Hong. "The Design of an Acquisitional Query Processor For Sensor Networks". *Proceedings of the 2003 SIGMOD Conference*, San Diego, CA, USA, June 2003, pp. 491-502.

[12] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, Wei Hong. "TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks". *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI 2002)*, Boston, MA, USA, December 2002.

[13] M. Maroti, B. Kusy, G. Simon, A. Ledeczi. "The Flooding Time Synchronization Protocol" *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SensSys 2004)*, Baltimore, MD, USA, November 2004, pp. 39-49.

[14] Matthew J. Miller, Nitin H. Vaidya. "Minimizing Energy Consumption in Sensor Networks Using a Wakeup Radio". *Proceedings of the Wireless Communications and Networking Conference 2004 (WCNC 2004)*, Atlanta, GA, USA, March 2004, pp. 2335-2340.

[15] Vijay Raghunathan, Curt Schurgers, Sung Park, Mani B. Srivastava. "Energy-Aware Wireless Microsensor Networks". *IEEE Signal Processing Magazine*, 19(2), March 2002, pp. 40-50.

[16] Curt Schurgers, Vlasios Tsiatsis, Saurabh Ganeriwal, Mani Srivastava. "Topology Management for Sensor Networks: Exploiting Latency and Density". *Proceedings of the 3rd ACM International Symposium on*

*Mobile Ad Hoc Networking and Computing (Mobihoc 2002)*, Lausanne, Switzerland, June 2002, pp 135-145.

[17] Curt Schurgers, Vlasios Tsiatsis, Saurabh Ganeriwal, Mani Srivastava. "Optimizing Sensor Networks in the Energy-Latency-Density Design Space". *IEEE Transactions on Mobile Computing (TMS)*, 1(1), January-March 2002, pp. 70-80.

[18] Yu-Chee Tseng, Chih-Shun Hsu, Ten-Yueng Hsieh. "Power-Saving Protocols for IEEE 802.11-Based Multi-Hop Ad Hoc Networks". *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 2002)*, New York, NY, USA, June 2002, pp. 200-209.

[19] Tijs van Dam, Koen Langendoen. "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks". *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, USA, November 2003, pp. 171-180.

[20] Ya Xu, John Heidemann, Deborah Estrin. "Geography Informed Energy Conservation for Ad Hoc Routing". *Proceedings of the 7th International Conference on Mobile Computing and Networking (MobiCom 2001)*, Rome, Italy, July 2001, pp. 70-84.

[21] Yong Yao, Johannes Gehrke. "The Cougar Approach to In-Network Query Processing in Sensor Networks". *SIGMOD Record*, 31(3), September 2002, pp. 9-18.

[22] Wei Ye, John Heidemann, Deborah Estrin. "An Energy-Efficient MAC Protocol for Wireless Sensor Networks". *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 2002)*, New York, NY, USA, June 2002, pp. 1567-1576.

[23] Wei Ye, John Heidemann, Deborah Estrin. "Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks". *IEEE/ACM Transactions on Networking*, 12(3), June 2004, pp. 493-506.

[24] Rong Zheng, Robin Kravets. "On-demand Power Management for Ad Hoc Networks". *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 2003)*, San Francisco, CA, USA, March-April 2003, pp. 481-491.