Silence is Golden and Time is Money: Power-Aware Communication for Sensor Networks

Christina Fragouli
École Polytechnique Fédérale de Lausanne
christina.fragouli@epfl.ch

Alon Orlitsky
University of California at San Diego
alon@ucsd.edu

Abstract

We propose protocols for energy-efficient communication over wireless sensor networks based on the use of silence as a means of conveying information. That is, information is inferred by the fact that nodes remain silent. We investigate the time-complexity trade-off such protocols offer, in a communication-complexity framework. We focus our attention on symmetric functions, that include most statistical functions and functions of interest for sensor networks.

1 Introduction

Silence has long been fabled to convey information. It is said that the Queen of X, once summoned all the women of her Kingdom and proclaimed: “It has come to my attention that not all your husbands are faithful. I was further told that each of you knows of all cheating husbands, except your own. I don't want you to discuss this sordid matter with anyone, but I do want you to ponder it long and hard. And should any of you some day determine for certain that your husband cheats, shoot him that very midnight.” And so left the Queen. Thirty nine tense but quite midnights went by, but on the fortieth night, forty gunshots were heard, and all forty cheating husbands of X ceased to exist. And thus, the women of old X used silence in lieu of communication to determine who cheated[1].

The use of time to communicate information dates back even further, and is even more substantiated. Fireflies are known to convey information using the intervals between light pulses. And German ethologist Karl von Frisch won the 1973 Nobel Prize for Physiology or Medicine for discovering that bees communicate distance to food sources via the duration of the waggle dance [3, 4].

[1]Or was it female intuition, see appendix.

Recently, another reason for using silence and timing to convey information has emerged. Sensor networks consist of small devices that gather and communicate information. Typically, a central processor, which we shall call a satellite, collects all that information and computes some function of the joint inputs. For example, the sensors may measure the temperature at various locations, and the satellite may determine their average, or whether some temperature exceeds a prescribed threshold.

Often, the sensors are small, have tiny batteries, and are not easily accessible. It is therefore desirable to find communication protocols that minimize their transmissions and thereby conserve their energy. We consider the use of silence and time to minimize the amount of communication required to compute various functions.

We focus on symmetric functions, which are functions that are invariant under permutation of their input arguments. Many common functions that might be of practical use in sensor-networks, such as average, max, and threshold are symmetric. Moreover, in the context of sensor networks, symmetric functions express the fact that the values of the measurements, rather than the identity of the sensors, is of importance.

To formalize silence, we consider *pulse communication* where at each time unit a node can either be silent, or emit an energy pulse. Unlike standard *bit communication*, where a node transmits either zero or one, the pulse itself conveys no information except for its existence. It can be thought of as a beam of light that either does or does not exist. It is the number of such pulses that we seek to minimize.

In Section 3, we consider the easiest functions to analyze, those whose inputs are binary. We consider protocols where one node transmits at a time. We show that in bit communication all non-constant functions require n bits in the worst-case. We then provide a simple characterization for the number of transmissions required for pulse communication of symmetric functions, showing for example that computing the n-variable OR and AND functions requires at most one pulse.

These results, for both bit and pulse communication come at a delay equal to n. For functions of variables over non-binary alphabets, pulse communication may result in higher delay than bit communication. For some applications time is of an essence, and we therefore also consider the tradeoff between communication and delay. In Section 4, we consider two scenarios. One where multiple nodes can communicate simultaneously. We show that the amount of communication C needed when delay $d < n$ is allowed satisfies the simple rule

$$C \sim A + \frac{n}{d},$$

where A is the number of pulses needed when one processor transmits at a time.

Analyzing communication complexity for sensor networks has been considered in [5], where asymptotic results for binary communication were derived. Our work differs in that, we consider pulse communication, use of silence, and calculate exact as opposed to asymptotic results. Our work can also be put in the framework of unit cost communication [6]. A new ingredient that we bring is that, unlike previous work we do not consider point-to-point communication, but instead distributed communication over the nodes of the sensor network.

The paper is organized as follows. Section 2 introduces our model and notation. Section 3 derives the complexity of pulse communication for binary-input symmetric functions. Section 4 investigates complexity – delay trade-offs, and Section 5 concludes the paper.

2 The communication model

We assume a very simple communication model. We consider a sensor network with n nodes that collect a set of measurements $\overline{x} = \{x_1,\ x_2\ \ldots, x_n\}$. Node i observes a value x_i from a discrete alphabet X_i. A central processor, which we shall call a satellite, would like to compute a function f of the joint inputs

$$f:\ X_1 \times X_2 \times \ldots \times X_n \to Z. \tag{1}$$

We assume that time is divided into slots of equal duration. The nodes follow an agreed-upon communication protocol. At each time slot at most one node transmits. Which node may transmit at a given time is determined by previous transmissions. What the node transmits is determined by previous transmissions and its own value. Note that for the transmissions to depend on previous ones, either the satellite polls the nodes, at each time asking a specific one for a function (determined by previous transmissions) of its value, or all nodes listen to all transmissions.[2]

One can consider two types of communication. Standard *bit communication* where whenever a node is selected to transmit it must transmit either 0 or 1. It is not allowed to remain silent. Mostly however, we will consider, *pulse communication* where at each time one node is selected to transmit, and it can either transmit or not. Transmission does not convey value, just the absence of silence. One way to visualize this is to think of the node as either shining a light, or not. If a light is shined, it does not convey a value, such as 0 or 1. These concepts are illustrated by the following example.

Example 1. Suppose that each of n nodes holds a binary value, and they want to determine the OR of these values, namely whether at least one of them has a 1.

A possible protocol is to agree on an ordering of the nodes from 1 to n. Then at the ith time slot, if node i has a 0 it remains silent, and if it has a 1 it emits a pulse. Once one node

[2]Although not examined in this paper, our approach can be extended in the case where nodes communicate through multiple hops.

emits a pulse, the communication stops, and the value of the function is determined to be 1. If after n time slots no node emits a pulse, the value of the function is 0. Note that the number of pulses emitted by all nodes is either zero or one. $\square$

We assume that the power consumed by the nodes to listen to ambient communication and to perform calculations is negligible. The only power measured is that required to transmit, and it is this power that we want to minimize. We will evaluate our protocols using two metrics. The maximum number of transmissions $C(f)$ the protocol might require, and the maximum number of time slots $T(f)$ we will need to evaluate the function f. In other words, we will calculate the worst-case complexity and delay required. In the traditional approach, where during each time slot one node transmits a binary value, it always holds that $T(f) = C(f)$. Thus, traditionally, only $C(f)$ is calculated. Using our approach, we will have that in general

$$T(f) \geq C(f). \tag{2}$$

We will restrict our attention to the case where nodes observe values over the same discrete alphabet of size k, that is, $X_i = [k] = \{1 \ldots k\}$, for each i. This is a natural assumption for a sensor network where sensor nodes are identical devices, as is almost always the case.

Moreover, we will consider functions f that are symmetric. A function f is called symmetric, if

$$f(x_1, x_2, \ldots, x_n) = f(\sigma(x_1, x_2 \ldots, x_n)), \tag{3}$$

where σ is any permutation of the functions arguments. The value of a symmetric function depends only on the histogram of the values of its arguments. This set of functions includes most statistical functions, such as the mean and the max value, and corresponds to sensor network applications where the measurements rather than the identity of the nodes is important. In the following we summarize our notation:

n : Number of nodes that observe values $x_1, \ldots, x_n$.

k : Each node observes a value in the set $X_i = [k] = \{1 \ldots k\}$.

f : A function of $x_1, \ldots, x_n$ that we want to evaluate.

$C(f)$: The (worst case) number of required transmissions (complexity).

$T(f)$: The (worst case) number of required time-slots (time).

3 Binary variables

Functions of binary variables are particularly easy to analyze. A symmetric function of binary inputs is determined by

$$|\overline{x}| = |\{i : x_i = 1\}|,$$

the number of ones among the inputs. A symmetric function f can thus be expressed as a function of its argument's weight,

$$f(\overline{x}) = f_I(|\overline{x}|).$$

For example, for the OR function,

$$f_I(|\overline{x}|) = \begin{cases} 1 & |\overline{x}| \geq 1, \\ 0 & |\overline{x}| = 0. \end{cases}$$

3.1 Bit communication

We first consider the standard bit communication model where each sensor transmits its value. The results are predictable and provided only to motivate and compare with the pulse communication model. Clearly, if the function is constant, namely attains the same value for all inputs, then no transmissions are necessary. We show that in all other cases, the worst-case communication is n.

Lemma 1. For every non-constant n binary-variable functions f,

$$C(f) = n, \quad T(f) = n.$$

Proof If f is non-constant, then $f_I(\alpha) \neq f_I(\alpha + 1)$ for some $0 \leq \alpha < n$. For $j = 1, \ldots, n$, let j be the jth sensor to transmit its value x_j. Consider the input $\overline{x} = \{x_1,\ x_2\ \ldots, x_n\}$ where

$$x_j = \begin{cases} 1 & j \leq \alpha, \\ 0 & j > \alpha. \end{cases}$$

It is easy to see that for this input, all processors must transmit to determine the functions value. □

Example 2. For the OR function, the α described in the previous lemma is 0. If $\overline{x} = \overline{0}$, all processors must transmit.

For the AND function, the α described in the previous lemma is $n - 1$. If the first $n - 1$ processors to transmit upon hearing all 1's in previous transmissions are assigned a 1, and the last to transmit in that case is assigned a 0, then all processors must transmit. □

3.2 Pulse communication

We now consider pulse communication, where a node that transmits a pulse does not convey any other information apart from the fact that it chose to transmit.

An *interval* is a collection of consecutive integers. For a given function f, an interval in $\{0, 1, \ldots, n\}$ is f-constant if f_I is constant for all values in the interval. Let $I_{\max}(f)$ be the cardinality of the largest f-constant interval.

Lemma 2. For all symmetric functions f of n boolean variables,

$$C(f) = n + 1 - I_{\max}(f), \quad T(f) = n. \tag{4}$$

Proof We can depict a symmetric function of binary variables as in Fig. 1, where the y-axis represents $|\overline{x}|$ (the number of ones in the input) and the x-axis represents $n - |\overline{x}|$ (the number of zeros in the input). Note that, if for given input we know the exact value of y, then we also know the value of x since $x + y = n$. The value of the function is depicted on the line $x + y = n$.

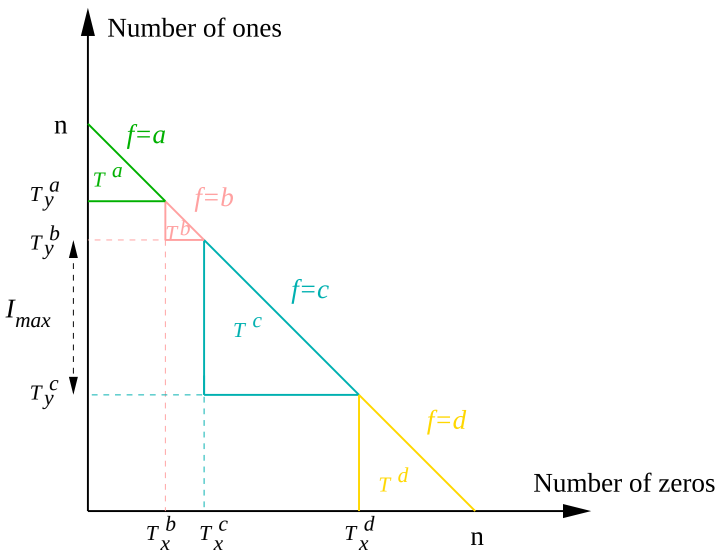

Figure 1: A method to represent a binary-input symmetric function.

Each f-constant interval $I(f)$ corresponds to an isosceles triangle, with two equal sides of length $I(f)$. We will describe a protocol that calculates the value of the function f by determining in which f-constant interval $|\overline{x}|$ belongs to. This protocol requires in the worst case $C(f) = n + 1 - I_{\max}(f)$ pulse transmissions, thus proving the achievability of Lemma 2.

Each node is allocated a time slot. At time slot i, node i will transmit or remain silent. We will keep a counter $S^i = (S_x^i, S_y^i)$ were S_x^i denotes the number of 0's, and S_y^i is the number of 1's, revealed up to and including time-slot i. Each value of S^i corresponds to a point on the plane, that expresses the information we have gathered thus far regarding the values of the input. Initially $(S_x^0, S_y^0) = (0, 0)$. During each time slot i, either S_x^{i-1} or S_y^{i-1} is increased by one, which visually corresponds to the point moving either right or up on the plane.

Note that, to uniquely determine the value of the function, we do not need to know the exact value of all arguments of the function: it is sufficient for the point S^i to belong to any of the triangles. The goal of the following argument is to achieve that using the minimum number of transmissions.

Protocol

Let T be the triangle corresponding to $I_{\max}(f)$ and let T_x and T_y denote the coordinates of T's base (namely its lower left corner), with $0 \leq T_x \leq n$, $0 \leq T_y \leq n$, and $T_x + T_y + I_{\max}(f) = n+1$. Nodes emit pulses if they have value 1, until time-slot m_1, where

• Time-slot $m_1 = n$ is reached, or,

• $m_1 < n$ but $S_y^{m_1} = T_y$, i.e., T_y nodes have emitted a pulse.

At this point, from the m_1 nodes that had the opportunity to transmit, $S_y^{m_1} = T_y$ have value 1, and $S_x^{m_1} = m_1 - T_y$ have value 0. Thus, we know that $T_y < |\overline{x}| \leq n - m_1$. If at time m_1 the value of the function is not uniquely determined, then from time $m_1 + 1$ and on, nodes emit pulses when they have value 0, until time-slot m_2, where

• Time-slot $m_2 = n$ is reached, or,

• $m_2 < n$ but $S_x^{m_2} = T_x$. In this case, since $S_y^{m_2} \geq S_y^{m_1} \geq T_y$ and $S_x^{m_2} \geq T_x$, the point S^{m_2} belongs to the "largest" isosceles triangle, and the value of the function is uniquely determined.

In the worst case, at most $T_y + T_x = n + 1 - I_{\max}(f)$ nodes will transmit.

We further need to show that there does not exist a protocol, that uses the same communication model, and results in a smaller worst case complexity. Consider any protocol and an input such that $|\overline{x}| \in I_{\max}(f)$. For the protocol to determine the value of the function, we need to learn the value of at least $n + 1 - I_{\max}(f)$ nodes. For any ordering of the nodes, there exists a worst case input that requires that many pulse transmissions. $\square$

Example 3. For $0 \leq \alpha \leq n+1$, the *greater than or equal* α function,

$$\mathrm{ge}_\alpha(\overline{x}) \stackrel{\text{def}}{=} \begin{cases} 1 & |\overline{x}| \geq \alpha, \\ 0 & |\overline{x}| < \alpha, \end{cases}$$

is 1 iff at least α of the n inputs are 1. For example, ge_0 is the constant-1 function, ge_1 is OR, $\mathrm{ge}_{n/2}$ is majority, ge_n is AND, and ge_{n+1} is the constant-0 function.

For $\alpha = 0$ or $n+1$, ge_α has one segment, $\{0, \ldots, n\}$, hence $I_{\max} = n+1$, implying that $C = 0$. Since these functions are constant, they require no communication.

For $1 \leq \alpha \leq n$, ge_α has two segments, $\{0, \ldots, \alpha - 1\}$, and $\{\alpha, \ldots, n\}$, hence $I_{\max} = \max(\alpha, n+1-\alpha)$, implying that $C = \min(\alpha, n+1-\alpha)$. The protocol that achieves this complexity is very simple. Assume wolog that $\alpha \leq (n+1)/2$. The processors agree on some order, and each one in its turn pulses if it has value one and keeps quite otherwise. Once α processors pulse, or all of them had a chance to communicate, communication stops. If $\alpha > (n+1)/2$, we would have instead nodes that have value zero emmit a pulse. $\square$

Example 4. For $0 < \alpha < n$, the *equal* α function,

$$\mathrm{eq}_\alpha(\overline{x}) \stackrel{\mathrm{def}}{=} \begin{cases} 1 & |\overline{x}| = \alpha, \\ 0 & |\overline{x}| \neq \alpha, \end{cases}$$

is 1 iff exactly α of the n inputs are 1.

Then eq_α has three constant segments, $\{0,\dots,\alpha-1\}$, $\{\alpha\}$, and $\{\alpha+1,\dots,n\}$, hence $I_{\max} = \max(\alpha, n-\alpha)$, and $C = \min(\alpha+1, n+1-\alpha)$. The same simple protocol achieves this number of bits, where if $\alpha \leq (n+1)/2$, nodes that have value one emit a pulse up to at most C pulses. □

Example 5. For the Exclusive-Or function,

$$\mathrm{xor}(\overline{x}) = (|\overline{x}|)_2,$$

each of the $n+1$ singletons in $\{0,\dots,n\}$ is a constant interval, hence $I_{\max} = 1$. It follows that $C = n$, namely all processors must transmit. □

Example 6. For an integer $1 \leq \alpha \leq n$, the *approximate sum* function

$$\mathrm{sum}_\alpha(\overline{x}) \stackrel{\mathrm{def}}{=} \left\lfloor \frac{\alpha|\overline{x}|}{n+2} \right\rfloor$$

approximates the number of 1's into α groups. For example, sum_2 is majority and sum_n is the number of ones. Then $I_{\max} = \lceil n/\alpha \rceil$, and therefore $C = n + 1 - \lceil n/\alpha \rceil$. It is easy to see that if as many processors emit a pulse, we know the value of $\mathrm{sum}_\alpha(\overline{x})$. □

4 Power-delay tradeoffs

The protocols considered thus far use silence to minimize power consumption. However, the resulting computation delay can reach $(k-1)n$, as is the case in the following example.

Example 7. Consider the "threshold" function f with ternary inputs, i.e., $k = \{0, 1, 2\}$.

$$f(\overline{x}) = \begin{cases} 1 & w_0 \geq \alpha \text{ and } w_1 \geq \beta \text{ and } w_2 \geq \gamma \\ 0 & \text{otherwise,} \end{cases}$$

where w_i is the number of i values the sensors observe. Assume that the values α, β, γ are such that $\alpha + \beta + \gamma << n$. Following a protocol similar to the binary-input case, we can first have up to α nodes that have observed value 1 transmit a pulse, then up to β nodes that observe value 2 transmit, and finally up to γ nodes that observe value 3 transmit. Thus the worst case complexity will be $C = \alpha + \beta + \gamma$. It is easy to see that the worst case delay will be $T = 2n$. One way to think about it is, that each of the n nodes will need two slots to convey, using one pulse, which of the three values it has. □

For some applications, time/delay may also be of essence. An underlying assumption of the discussed protocols is that the each node is allocated one time-slot and emits at most one pulse, basing its possible transmission on those of previous nodes. If computation must be performed more rapidly, one can relax this constraint and decrease the delay at the cost of increased power consumption. We now consider simultaneous node transmission where multiple nodes transmit at the same time, and multi-pulse encodings where nodes use more than one pulse to encode their values.

4.1 Simultaneous transmissions

In this model nodes are allowed to simultaneously transmit during the same time slot. We assume that when this happens, the satellite knows how many nodes transmitted. This is a reasonable assumption, because, transmissions are over a unary alphabet, and thus, the satellite can simply measure the level of received energy to determine how many nodes have simultaneously transmitted during a time-slot. To visualize this mode of communication, one can imagine some subset of the sensors shining what may be a thousand points of light, and the satellite observing which of the nodes emit light.

In particular, we can divide the n nodes into d sets of size at most $\lceil n/d \rceil$ each, and allocate a time-slot to each set. During each time slot, all nodes in the corresponding set are allowed to possibly emit a pulse. Lemma 2 corresponds to the case where $d = n$, and can be easily generalized as follows.

Lemma 3. For all symmetric functions of n boolean variables, we can achieve

$$T = d, \quad C \leq n + \lceil n/d \rceil - I_{\max}.$$

□

4.2 Multi-pulse encodings

For functions over k-size alphabets, each node may take $k-1$ time slots to convey its value. For binary alphabets, this delay is 1. For larger alphabets, the delay is larger. This delay can be reduced by using multi-pulse encodings.

Let node i observe a value x_i in $[k]$. To convey this value we can achieve the following complexity-time trade-off.

• *One-pulse encoding:* $T = k - 1, \quad C = 1.$

Allocate $k-1$ time slots to the node. By not transmitting, or transmitting one pulse at one of the $k-1$ time-slots, the node can convey k values.

• *Binary encoding:* $T = \log k, \quad C = \log k.$

Viewing silences as 0 and pulses as 1, we can convey the value in $\lceil \log k \rceil$ time slots. However, the number of pulses can be as large as the number of slots.

• *Multi-pulse encoding:* $T = t \approx ck^{\frac{1}{c}}$, and $C = c$.
We can use

$$\binom{t}{c} + \binom{t}{c-1} + \ldots + \binom{t}{1} + 1, \tag{5}$$

distinct values. We are interested in the case where the number of transmissions c is smaller than $\frac{t}{2}$ - otherwise we will get that $t \approx c \approx \log k$ and the resulting complexity will resemble the binary encoding case. For $c << t$ we can approximate the sum in Eq. (5) as $\binom{t}{c}$ and thus

$$\binom{t}{c} \approx 2^{th(\frac{c}{t})} \approx 2^{c \log \frac{t}{c}} \approx k.$$

5 Conclusions

We proposed protocols that use silence and timing to convey information in an energy efficient manner over wireless sensor networks. We characterized the worst case complexity required for binary-input symmetric functions, and investigated delay-complexity trade-offs possible over higher alphabet values.

References

[1] A. Orlitsky and A. El Gamal, "Communication complexity", Complexity in Information Theory, Y. Abu Mustafa (editor), Springer-Verlag, 1986, pp. 16-61.

[2] Eyal Kushilevitz and Noam Nisan "Communication Complexity", Cambridge Univ. Press, 1996.

[3] Karl Von Frisch, "Dancing Bees: An Account of the Life and Senses of the Honey Bee", http://www.polarization.com/bees/bees.html

[4] "The flight paths of honeybees recruited by the waggle dance", Nature 435, May 2005, pp.205-207.

[5] A. Giridhar and P. R. Kumar, "Computing and communicating functions over sensor networks", IEEE Jouranl on Selected Areas in Communications, pp. 755-764, vol. 23, no. 4, April 2005.

[6] S. Verdu, "On Channel Capacity per Unit Cost", IEEE Trans. Information Theory, vol. IT-36, no. 5, pp. 1019-1030, Sep. 1990.