Fundamental Protocols for Wireless Sensor Networks[*]

Raghuvel S. Bhuvaneswaran, Jacir L. Bordim, Jiangtao Cui, and Koji Nakano
Department of Electrical and Computer Engineering
Nagoya Institute of Technology,
Showa-ku, Nagoya 466-8555, Japan
(bhuvan,jace,cjto,nakano)@maple.elcom.nitech.ac.jp

Abstract

The main contribution of this work is to present energy-efficient protocols that compute the sum of n numbers over any commutative and associative binary operator stored in n wireless sensor nodes arranged in a two-dimensional grid of size $\sqrt{n} \times \sqrt{n}$. We first present a protocol that computes the sum in $O(r^2 + (\frac{n}{r^2})^{\frac{1}{3}})$ time slots with no sensor node being awake for more than $O(1)$ time slots, where r is the transmission range of the sensor nodes. We then show a fault-tolerant protocol that computes the sum in the same number of time slots with no sensor node being awake for more than $O(\log r)$ time slots.

1. Introduction

A Wireless Sensor Network (WSN for short) is a distributed system consisting of a base station and a number of wireless sensors nodes endowed with radio transceivers. The data being sensed by the sensor nodes in the network is eventually transmitted to a base station, where the information can be accessed. Figure 1 depicts a network of sensor nodes and a base station. Remote object monitoring and tracking is important in several different contexts, such as traffic management, production-line, detection of the presence or absence of certain objects, and so on. WSNs greatly extend the ability of monitoring and controlling the physical environment from remote locations. If the sensors are empowered with the ability of sharing their observations and coordinating among themselves to gather and process information, then meaningful and useful data can be available at the base station. Users can retrieve information from the base station to control the environment from afar. Therefore, we envision a future in which collections of sensor nodes will be employed to form ad-hoc sensor networks.

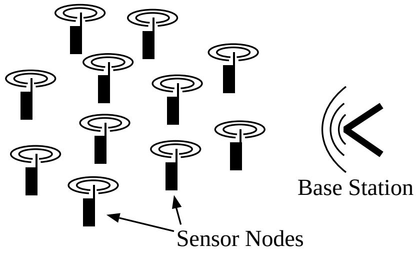


Figure 1. *A network of wireless sensor nodes and a base station.*

In a *single-hop* WSN, a sensor node can directly communicate with any other sensor node. In *multi-hop* WSN, however, the communication between two sensor nodes may involve a sequence of hops through a chain of pairwise adjacent sensor nodes. There is a single-hop communication between the base station and the sensor nodes, while the communication among the sensor nodes can be either single or multi-hop. There are several possible models for WSNs, in this work we consider WSNs where all the sensor nodes in the network are fixed, homogeneous and energy-constrained.

We assume that the base station and all the sensor nodes have a local clock that keeps synchronous time, perhaps by interfacing with the base station or with a GPS system [11]. All sensor nodes run the same protocol and can perform computations on the data being sensed. As customary, time is assumed to be slotted and all transmissions take place at slotted boundaries [3, 5]. It is assumed that at any time slot, a sensor node can communicate with the base station. The size of a data packet is such that its transmission can be completed within one time slot.

We employ the commonly-accepted assumption that when two or more sensor nodes are in the transmission range of each other and transmitting in the same time slot, the corresponding packets *collide* and are garbled beyond

[*] Work supported, in part, by Grant-in-Aid for Encouragement of Young Scientists (12780213) from Ministry of Education, Science, Sports, and Culture of Japan.

recognition. Similarly, when two or more sensor nodes are broadcasting a packet at the same time slot, the base station can not receive these packets.

The sensor nodes in the WSN are assumed to run on batteries and saving power is exceedingly important as recharging them may not be possible while in operation. On the other hand, the base station is not energy-constrained and hence power saving is not crucial for it. It is well known that a sensor node expends power while its transceiver is active, that is, while transmitting or receiving a packet. It is perhaps surprising at first that a sensor node expends power even if it receives a packet that is not destined for it. Accordingly, we assess the efficiency of a protocol by the following two metrics:

- The overall number of time slots required by the protocol to terminate.

- For each individual sensor node, the total number of time slots it has to be awake to transmit/receive packets.

The goal of optimizing these parameters are, of course, conflicting. It is relatively straightforward to minimize overall completion time at the expense of energy consumption. Similarly, one can minimize energy consumption at the expense of completion time. The challenge is to strike a sensible balance between the two by designing protocols that take a small number of time slots to terminate while keeping as energy-efficient as possible.

As mentioned earlier, one of the salient features of WSN is information gathering. In other words, the major task is to collect the information that is scattered among the sensor nodes. In this work, we developed a protocol to perform any commutative and associative binary operation over the values stored in the sensor nodes. The binary operation involves addition, multiplication, logical AND/OR, and so on. Our protocol performs the operations in coordination with base station by efficiently avoiding possible collisions.

This paper is organized as follows: Section 2 formally defines the problem and model. Section 3 provides some preliminary results used in the subsequent section. In Section 4, we present an algorithm to solve the binary operation in multi-hop WSNs and Section 5 concludes this work.

2 Model and Problem Definition

The base station is equipped with powerful batteries and has a large antenna which covers a wide area, so that the it can monitor all the sensor nodes under consideration. The base station and the sensor nodes can directly communicate. The computation among the sensors is performed in coordination with the base station. A sensor node (*sensor*, for short) in a single-hop WSN can tune to a channel to send/receive a packet. At the end of a time slot, the status of the channel is:

NULL: no packet has been driven into the channel in the current time slot;

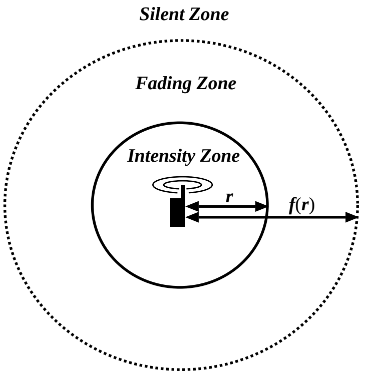
SINGLE: exactly one packet has been driven into the channel in the current time slot;

COLLISION: two or more packets have been driven into the channel in the current time slot.

Suppose that a sensor is positioned in a two dimensional plane. When a sensor transmits a packet with power r, the signal will be strong enough for other sensors to hear it within the Euclidean distance r from the sensor that originates the packet. In other words, to cover a range of r, the sensor that originates the signal must transmit with enough power to cover that range.

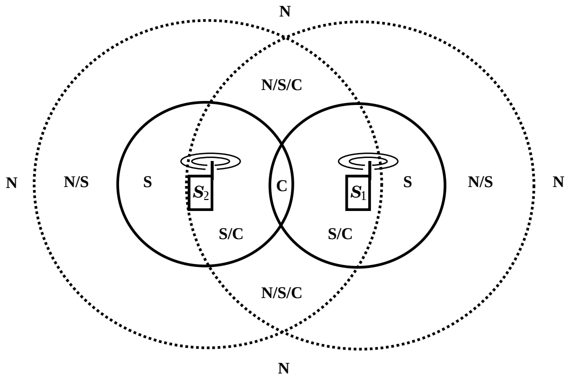
Every sensor in the *intensity zone*, that is, the region within the distance r from a sensor that originates the packet, is guaranteed to receive it. It is well known that signals are subject to fluctuations and start fading after traveling some distance [9]. Hence, those sensors outside the transmission range r of a source node, say $r + \delta$ for some $\delta > 0$, may or may not receive the packet. We formalize this situation as follows: the *fading zone* of a sensor is defined as the region outside the *intensity zone* and inside $f(r)$, where f is an increasing function. Clearly, those sensors in the *fading zone*, may or may not receive the packet. The status of the channel within the *intensity zone* is always SINGLE, and in the *fading zone*, it is either SINGLE or NULL. The sensors in the *silent zone*, that is, beyond the Euclidean distance $f(r)$ from the sensor that originated the broadcast are guaranteed not to receive the packet, and the status of the channel is always NULL. The status of the channel during a broadcast is illustrated in Figure 2. For simplicity, we assume that $f(r) = 2r$ and design the protocol under this assumption. Our protocol also works for a general function f as long as $f(r) = c \cdot r + o(r)$ for any fixed $c \geq 1$ by adjusting some parameters used in the protocol.

Suppose that two sensors S_1 and S_2 broadcast on the channel in the same time slot as illustrated in Figure 3. When the transmissions of S_1 and S_2 overlap, a *collision* occurs and any sensor that lies between them cannot correctly receive the packets. The sensors in the *intensity zone* of S_1 and in the *fading zone* of S_2 may or may not receive the packet from S_1 due to the interference caused by the signal originated from S_2. In this case, the status of the channel is either SINGLE or COLLISION. Similarly, sensors in the *intensity zones* of both S_1 and S_2 cannot receive packets and the status of the channel is always COLLISION. Sensors in the *fading zone* of S_1 and S_2 may or may not receive a packet from S_1 as well as S_2. The status of the channel, in this case, is NULL, SINGLE, or COLLISION. The



Zone	Channel Status
Silent Zone	NULL
Fading Zone	NULL or SINGLE
Intensity Zone	SINGLE

Figure 2. *Transmission zones of a sensor node.*



Channel Status	
N	NULL
S	SINGLE
C	COLLISION

Figure 3. *Status of the channel when the transmissions of two sensor nodes overlap.*

reader can easily generalize the above facts to three or more sensors.

In this work, we assume that the sensor nodes in the WSN are organized as a two dimensional square plane of size $\sqrt{n} \times \sqrt{n}$ with coordinates (x, y), $(1 \leq x, y \leq \sqrt{n})$. The plane can be viewed as n cells of unit size 1×1. Let $C(x, y)$, $(1 \leq x, y \leq \sqrt{n})$, denote a cell consisting of all points (x', y'), $(x \leq x' < x + 1; y \leq y' < y + 1)$. Suppose that each cell $C(x, y)$ has a sensor denoted $S_{x,y}$. Throughout this work we assume that each sensor node $S_{i,j}$, $(1 \leq i, j \leq \sqrt{n})$, knows its cell location within the grid. Clearly, for any two sensors located in adjacent cells, the farthest distance between them is $\sqrt{5}$ as illustrated in Figure 4. Hence, to ensure the communication between adjacent sensors, a packet must be transmitted with power of at least $\sqrt{5}$ to cover a region of $\sqrt{5} \approx 2.24$. Similarly, sensors in diagonal adjacent cells have distance of at most $2\sqrt{2}$. Thus, a sensor has to transmit with enough power to cover an area of at least $2\sqrt{2} \approx 2.83$ to ensure communication with its neighbors. If the sensors on a WSN of size $\sqrt{n} \times \sqrt{n}$ can broadcast with sufficient power to cover an area of $\sqrt{2n}$, then, any pair of sensors can directly communicate. In such case, the WSN essentially allows a single-hop communication. In other words, if the sensors with transmission range r are allocated on the WSN of size $\frac{\sqrt{2}r}{2} \times \frac{\sqrt{2}r}{2}$ ($\approx 0.71r \times 0.71r$), then a single-hop communication is ensured.

In the WSNs, sensor nodes may fail due to power dissipation, intentional or unintentional damages and so on. Therefore, we have devised a protocol that will work even in the presence of faulty sensor nodes. We assume that at least one sensor node remains active for a given group during the execution of the protocol.

Suppose that each sensor $S_{i,j}$, $(1 \leq i, j \leq \sqrt{n})$, has a value $x_{i,j}$, which could represent temperature, humidity, gravity, seismic information, etc. Let $\otimes$ be any commutative and associative binary operation such as addition, multiplication, or finding the maximum. The *sum* problem is to perform the $\otimes$ operation over all $x_{i,j}$, that is,

$$X = x_{1,1} \otimes x_{1,2} \otimes, \ldots, \otimes x_{\sqrt{n},\sqrt{n}}.$$

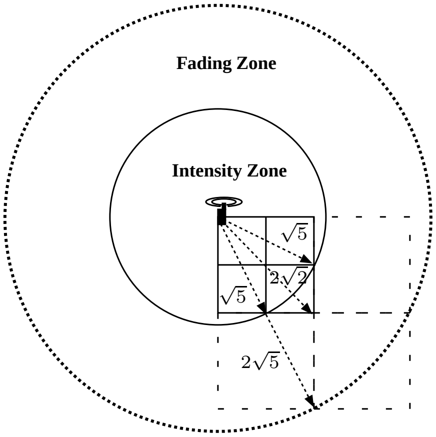


Figure 4. *A sensor node has to transmit with enough power to ensure communication with sensors within adjacent cells.*

In the next section, we present some protocols to solve the sum problem in single-hop WSNs. These results will be later used in developing a protocol for multi-hop WSNs.

3 Protocols to Collect Information Among the Sensor Nodes

Suppose that a WSN has m sensor nodes where all of them lie in the transmission range of each other and each one has a unique ID in $[1, m]$. Let S_i denote a sensor node with ID i $(1 \leq i \leq m)$, which has a value x_i stored in it.

The sum problem can be solved in $m - 1$ time slots as follows: For each time slot i $(1 \leq i \leq m-1)$ a sensor node S_i broadcasts x_i on the channel and sensor node S_{i+1} monitors the channel. After receiving the value x_i, the sensor node S_{i+1} computes $x_{i+1} = x_i \otimes x_{i+1}$. Clearly, this protocol takes $m-1$ time slots. This protocol is energy-efficient, since each sensor node is awake for at most 2 time slots. Thus, we have the following lemma,

Lemma 1 *The sum can be computed on a WSN with m sensor nodes, where each sensor node has a unique ID in the range $[1, m]$, in $m - 1$ time slots and each sensor node is awake for at most 2 time slots without involving the base station.*

Obviously, the above protocol is not fault tolerant. If any of the sensor nodes in the WSN fails, this protocol is unable to yield the correct result. To overcome this problem, we have devised a fault tolerant protocol. We assumed that faults will not occur during the execution of the protocol and at least one sensor node remains active in the WSN.

For each time slot i $(1 \leq i \leq m)$, sensor node S_i broadcasts x_i on the channel, and every sensor node S_k, $(k > i)$ monitors the channel to receive the value. After receiving x_i, each sensor node S_k computes $x_k = x_i \otimes x_k$. When a sensor node S_j $(j < i)$ hears the broadcast of S_i, it knows that S_i exists and leaves the protocol. Consequently, this protocol terminates in m time slots. The worst case occurs when a single sensor node remains active in the WSN. In such case, this sensor node will be awake for exactly m time slots, that is, the entire duration of the protocol. We summarize our findings in the following lemma.

Lemma 2 *Even in the presence of faulty sensor nodes, the sum can be computed on a WSN with m sensor nodes, where each sensor node has a unique ID in the range $[1, m]$, in m time slots and no sensor node is awake for more than m time slots without involving the base station.*

Although this protocol works in the presence of faulty sensor nodes, it is not energy-efficient. We now introduce a fault tolerant and energy-efficient protocol to compute the sum in a single-hop WSN. When the protocol terminates the following two conditions are satisfied:

(Cnd-1) The last active sensor node, denoted as S_k, such that no sensor node S_i with $i > k$ exists, has been identified and holds the final result.

(Cnd-2) The protocol takes $2m-2$ time slots and no sensor node is awake for more than $2 \log m$ time slots.

If $m = 1$, then S_1 knows x_1 and the above conditions are verified. Now, assume that $m \geq 2$. The m sensor nodes are partitioned into two groups $\mathcal{P}_1 = \{S_i \mid 1 \leq i \leq \frac{m}{2}\}$ and $\mathcal{P}_2 = \{S_i \mid \frac{m}{2}+1 \leq i \leq m\}$. Recursively compute the sum in $\mathcal{P}_1$ and $\mathcal{P}_2$. By the induction hypothesis, the above conditions **Cnd-1** and **Cnd-2** are satisfied and, therefore, each of the two subproblems can be solved in $m-2$ time slots, with no sensor node being awake for more than $2 \log m - 2$ time slots. Let S_j and S_k be the last active sensor nodes in groups $\mathcal{P}_1$ and in $\mathcal{P}_2$, respectively. In the next time slot, sensor node S_j transmits x_j, that is $\sum\{x_i \mid 1 \leq i \leq j$ and S_i exists$\}$, on the channel. The last active sensor node S_k in $\mathcal{P}_2$ monitors the channel and updates the result. In one additional time slot the sensor node S_k reveals its identity. The reader can easily confirm that the protocol satisfies the aforementioned conditions **Cnd-1** and **Cnd-2**. Thus, we have the following result,

Lemma 3 *Even in the presence of faulty sensor nodes, the sum can be computed on a WSN with m sensor nodes, where each sensor node has a unique ID in the range $[1, m]$, in $2m-2$ time slots with no sensor node being awake for more than $2 \log m$ time slots without involving the base station.*

4 Protocol for Multi-hop WSNs

In this section, a protocol to solve the sum problem is presented for multi-hop WSNs. A naive protocol can compute the sum in a WSN with n sensor nodes in $O(n)$ time slots as follows: each sensor node $S_{i,j}$, $(1 \leq i, j \leq \sqrt{n})$, broadcasts its value $x_{i,j}$ on the channel one at a time. The base station monitors the channel and computes the final result. Since there are n sensor nodes, and each sensor node transmits its value on a different time slot, the protocol takes n time slots to complete and each sensor node is awake for only one time slot.

Our goal is to minimize the overall completion time of the protocol while keeping the number of awakening time slots at minimum. Thus in the next sub-section we present a protocol that solves the sum problem in $O(r^2 + (\frac{n}{r^2})^{\frac{1}{3}})$ time slots.

4.1 Protocol to Solve the Binary Operation Problem

The key idea behind our protocol is to partition the grid into blocks and sub-blocks and then compute the sum on each sub-block. The partial results are combined such that the operations in some blocks can be parallelized while avoiding possible collisions. The protocol is spelled out as follows:

`Protocol BinOp;`

Step 1: The $\sqrt{n} \times \sqrt{n}$ cells are divided into $\frac{n}{k^2}$ groups with $k \times k$ cells each. Further, each group is partitioned into $\frac{k^2}{(\frac{9r}{4})^2}$ blocks with $\frac{9r}{4} \times \frac{9r}{4}$ cells each. Each block is further partitioned into 81 sub-blocks of size $\frac{r}{4} \times \frac{r}{4}$.

Step 2: Compute the sum on each sub-block $\frac{r}{4} \times \frac{r}{4}$.

Step 3: The sum on each block is obtained by combining the partial sums from **Step 2**.

Step 4: Compute the sum on each group by combining the results of **Step 3**.

Step 5: For each group, the sensor node that holds the sum of its group broadcasts it to the base station, one at a time. The base station monitors the channel and computes the overall sum.

The partitioning scheme of **Step 1** is illustrated in Figure 5. The grid has been partitioned in a way that the sensor nodes in a sub-block are within the transmission range of each other. Note that the maximum distance between any two sensor nodes of two neighboring sub-blocks is less than $\sqrt{(\frac{r}{4})^2 + (\frac{r}{2})^2} = \frac{\sqrt{5}}{4}r < r$. This guarantees that the sensor nodes in a sub-block are within the transmission range of each other and the sensor nodes within a sub-block can directly communicate with any sensor node on a neighboring sub-block. In other words, a single-hop communication is ensured between neighboring sub-blocks. Since the sensor nodes are within the transmission range of each other, the operations in **Step 2** can be computed using any of the protocols discussed in Section 3 by converting the sensor nodes' location into an ID number.

Let $S'_{i,j}$, $(1 \leq i \leq 9, 1 \leq j \leq 9)$, be the sensor node of each sub-block that holds the sum computed in **Step 2**. In **Step 3**, the sum of each block is computed by combining these partial results as shown in Figure 6. We begin by computing the sum on each row of each block, such that the sensor nodes located at the rightmost column of each block holds the sum of the row it belongs to. Next, the sum is performed only on the rightmost column of each block. Thus, at the end of **Step 3** the sensor node on the bottom right of each block holds the sum of its corresponding block. Let $S''_{i,j}$, where $(1 \leq i \leq \frac{k}{\frac{9r}{4}}, 1 \leq j \leq \frac{k}{\frac{9r}{4}})$, be the sensor node on the bottom right of each block that holds the sum of its corresponding block. Note that there are 8 sub-blocks of size $\frac{r}{4} \times \frac{r}{4}$ between $S''_{i,j}$ and $S''_{i,j+1}$. Clearly, these two sensor nodes cannot directly communicate because they are outside the transmission range of each other. Therefore, the sensor nodes in the sub-blocks that are between $S''_{i,j}$ and $S''_{i,j+1}$ will be used to forward packets from $S''_{i,j}$ to $S''_{i,j+1}$.

In **Step 4**, when a sensor node say, $S''_{1,j}$ broadcasts $x_{1,j}$, its right neighboring node in the sub-block (that is, $S'_{i,j}$) hears the broadcast and forwards it until the value reach the sensor node $S''_{1,j+1}$. The sensor node $S''_{1,j+1}$ updates its value and broadcasts it. This process is repeated until the sum of the entire row has been computed. Note that the above computation is performed in parallel for every row on each group as shown in Figure 7. The sum of the rightmost column of each group is computed in the same way.

In **Step 5**, the sensor node located at the bottom right corner of each group broadcasts the sum of its group to the base station, in turn. The base station monitors the channel and computes the overall sum as shown in Figure 8.

4.2 Analysis

Now let us discuss the time complexity of our protocol. Clearly, **Step 1** uses no broadcast. **Step 2** can be computed in $O(r^2)$ time slots using any of the protocols presented in Section 3. The number of awake time slots depends upon the protocol we choose, it can be either $O(1)$ or $O(\log r)$ according to Lemma 1 and Lemma 3, respectively. Note that the sum on each of the 81 sub-blocks of a block is performed sequentially. However, the sum on sub-blocks of neighboring blocks are performed in parallel.

Step 3 can be sub-divided into two phases: Phase 1, where the results of every row on each block is combined,

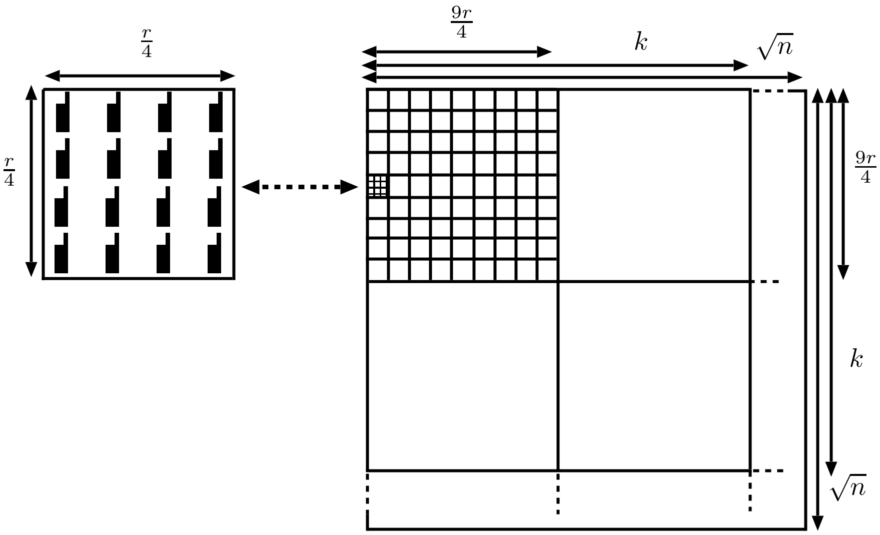


Figure 5. *The $\sqrt{n} \times \sqrt{n}$ grid is partitioned into sub-blocks.*

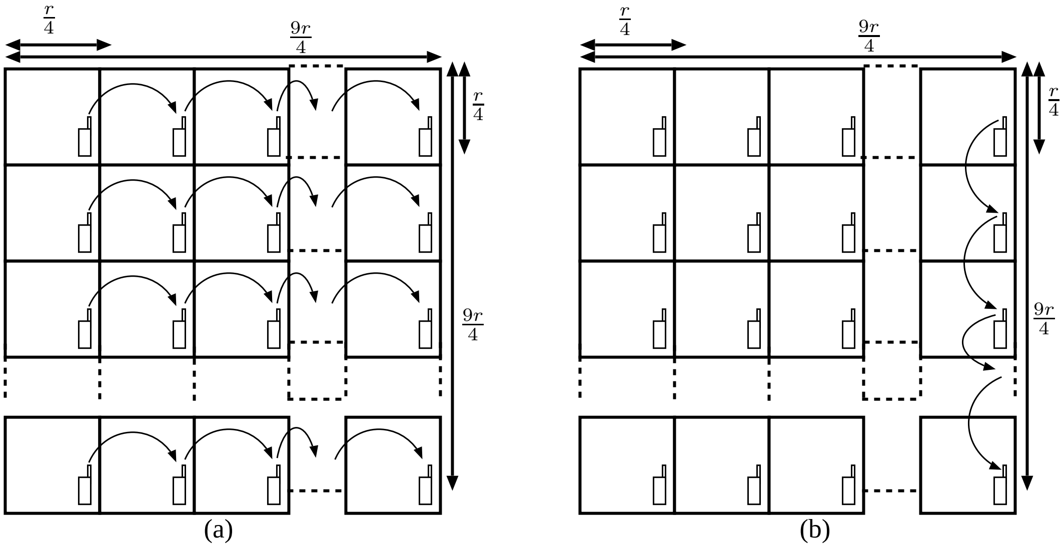


Figure 6. *(a) The binary operation is performed on each row sequentially. (b) The binary operation is performed on the rightmost column.*

and Phase 2, where the results of the rightmost column on each block is computed. In Phase 1, when a sensor node $S'_{i,j}$ broadcasts, only the sensor node $S'_{i,j+1}$ that lies on the right of $S'_{i,j}$, is awake to receive the packet. Therefore, to compute the sum on each row, each sensor node is awake for at most 2 time slots. In Phase 2, no sensor node needs to be awake for more than 2 time slots, and $S'_{9,9}$ is awake for only 1 time slot. Note that only those sensor nodes that were awaken for 1 time slot to receive the packet in Phase 1 will participate in Phase 2. Clearly, in **Step 3** no sensor node needs to awake for more than 3 time slots.

In **Step 4**, the sum of each row of each group is computed in parallel. Recall that the sensor node $S''_{i,j}$ can not directly communicate with the sensor node $S''_{i,j+1}$. The sensor nodes that lie between them forward the packets from $S''_{i,j}$ to $S''_{i,j+1}$. Thus, to compute the sum on the rows of all groups take $9 \times (\frac{k}{\frac{9r}{4}} - 1)$ time slots, and to compute the sum on the rightmost column incurs another $9 \times (\frac{k}{\frac{9r}{4}} - 1)$ time slots. Thus, **Step 4** can be computed in $O(\frac{k}{r})$ time slots and

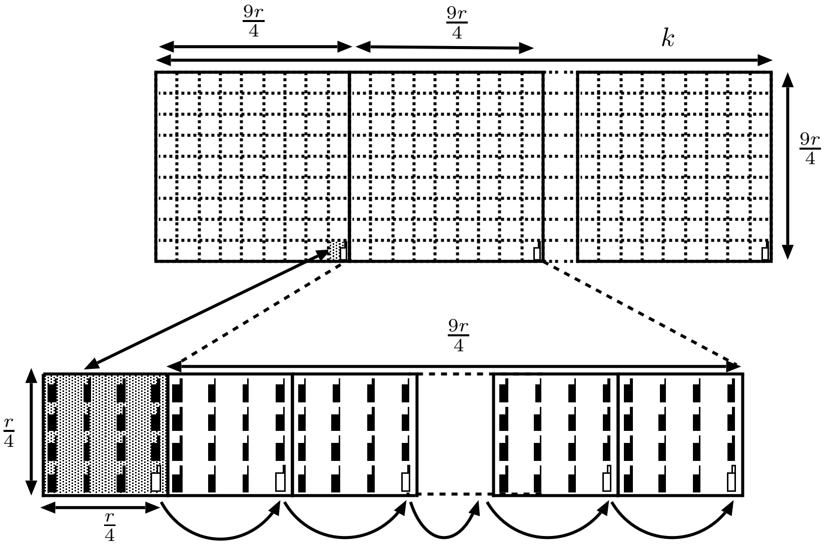


Figure 7. *The computation is performed in parallel for every row on each group.*

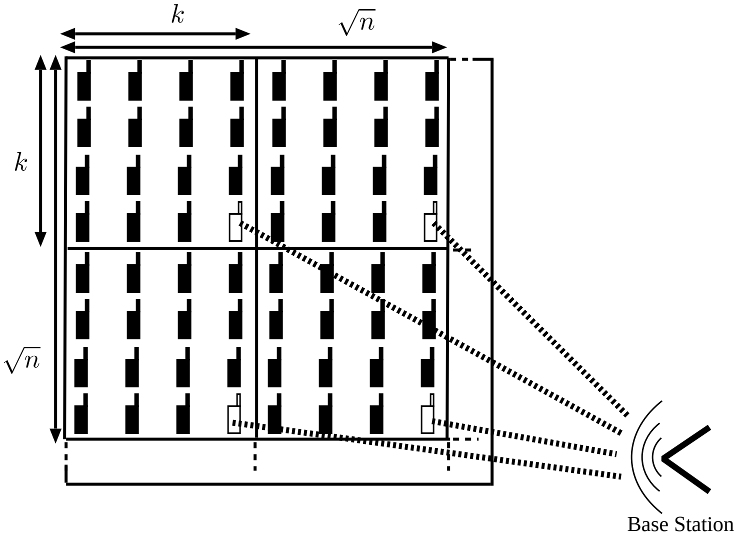


Figure 8. *The sensor nodes transmit the sum of its group to the base station.*

no sensor node needs to be awake for more than 3 time slots. Those sensor nodes that hold the sum of their group, broadcast to the base station in turn. Since there are $\frac{n}{k^2}$ groups, **Step 5** takes $O(\frac{n}{k^2})$ time slots to compute the sum and the sensor nodes participating in this step are awake for exactly 1 time slot.

Hence, our protocol takes $O\left(r^2 + \frac{k}{r} + \frac{n}{k^2}\right)$ time slots to compute sum of n numbers on a WSN. The time complexity of our protocol can be minimized by properly selecting the parameter k. With $k = (nr)^{\frac{1}{3}}$ the time complexity becomes $O(r^2+(\frac{n}{r^2})^{\frac{1}{3}})$. To summarize, we have the following result,

Lemma 4 *On a WSN where the sensor nodes are organized as a grid, the sum of n numbers can be computed in $O(r^2 + (\frac{n}{r^2})^{\frac{1}{3}})$ time slots when $r \gg 1$.*

Combining Lemma 1 and Lemma 4 we obtain the following corollary.

Corollary 1 *On a WSN where the sensor nodes are organized as a grid, the sum of n numbers can be computed by an energy-efficient protocol in $O(r^2 + (\frac{n}{r^2})^{\frac{1}{3}})$ time slots when $r \gg 1$, and no sensor needs to awake for more than $O(1)$ time slots.*

Combining the Lemma 3 and Lemma 4 we obtain the following corollary.

Corollary 2 *On a WSN where the sensor nodes are organized as a grid, the sum of n numbers can be computed by a fault-tolerant and energy-efficient protocol in $O(r^2 + (\frac{n}{r^2})^{\frac{1}{3}})$ time slots when $r \gg 1$, and no sensor needs to awake for more than $O(\log r)$ time slots.*

Furthermore, if each sensor node is able to adjust its transmission range before the execution of the protocol, then with $r^2 = (\frac{n}{r^2})^{\frac{1}{3}}$ that is $r = n^{\frac{1}{8}}$, the time complexity of the above protocol becomes $O(n^{\frac{1}{4}})$.

5 Conclusion

WSNs can greatly augment the ability to control and supervise the environment from distant locations. In this work, we have focused on the problem of solving commutative and associative binary operation over the values stored in n sensor nodes in a WSN.

We proposed energy-efficient protocols that compute the sum of n numbers over any commutative and associative binary operator stored in n wireless sensor nodes arranged in a two-dimensional grid of size $\sqrt{n} \times \sqrt{n}$. We presented a protocol that computes the sum in $O(r^2 + (\frac{n}{r^2})^{\frac{1}{3}})$ time slots with no sensor node being awake for more than $O(1)$ time slots, where r is the transmission range of the sensor nodes. Next, we introduced a fault-tolerant protocol that computes the sum in the same number of time slots with no sensor node being awake for more than $O(\log r)$ time slots. We have designed our protocol in such a way that collisions are properly avoided.

References

[1] N. Abramson, *Multiple Access Communications: Foundations for Emerging Technologies*, IEEE Press, New York, 1993.

[2] N. Abramson, *Multiple access in wireless digital networks*, Proceedings of the IEEE, 82, (1994), 1360–1370.

[3] R. Bar-Yehuda, O. Goldreich, and A. Itai, *Efficient emulation of single-hop radio network with collision detection on multi-hop radio network with no collision detection*, Distributed Computing, 5, (1991), 67–71.

[4] R.Bar-Yehada, O. Goldreich, and A. Itai, *On the time complexity of broadcast in radio networks: An exponential gap between determinism and randomization*, Journal of Computer and Systems Science,45,(1992),104-126.

[5] D. Bertzekas and R. Gallager, *Data Networks*, Second Edition, Prentice-Hall, 1992.

[6] R. Binder, N. Abramson, F. Kuo, A. Okinaka, and D. Wax, *ALOHA packet broadcasting – a retrospect*, AFIPS Conf Proceedings, May 1975, 203–216.

[7] U. Black, *Mobile and Wireless Networks*, Prentice Hall, Upper Saddle River, NJ, 1996.

[8] Chlamtac. I. and Kutten.S, *On Broadcasting in Radio Networks- Problem Analysis and Protocol Design*, IEEE translations on Communications, December,(1985),Vol COM-33, No.12.

[9] M. P. M. Hall and L. W. Barclay *Radio-wave Propagation*, IEE Electro Magnetic Wave Series 30, Peter Peregrinus Ltd., 1989.

[10] M. Hofri, *A Feedback-less Distributed Broadcast Algorithm for Multihop Radio Networks with Time-varying Structure*, 2nd ACM intr. MCPR Workshop, Rome(May 1987).

[11] E. D. Kaplan, *Understanding GPS: principles and applications*, Artech House, Greenwich, 1998.

[12] R. M. Metcalfe and D. R. Boggs, *Ethernet: distributed packet switching for local computer networks*, Communications of the ACM, 19, (1976), 395–404.

[13] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, 1995.

[14] K. Nakano, Optimal initializing algorithms for a reconfigurable mesh, *Journal of Parallel and Distributed Computing*,vol.24, pp.218-223, Jan.1995.

[15] D. E. Willard, *Log-logarithmic selection resolutions protocols in a multiple access channels*, SIAM Journal on Computing. Vol.15, 468-477,(1986).