Function Computation over Heterogeneous Wireless Sensor Networks

Xuanyu Cao\textsuperscript{1,2}, Xinbing Wang\textsuperscript{1,3}, Songwu Lu\textsuperscript{4}

\textsuperscript{1}Dept. of Electronic Engineering, Shanghai Jiaotong University, China
\textsuperscript{2}State Key Laboratory of Electronic Engineering, Shanghai Jiaotong University, China
\textsuperscript{3}National Mobile Communications Research Laboratory, Southeast University, China
\textsuperscript{4}Dept. of Computer Science, University of California, Los Angeles, CA

Email: \{apogne\mbox{,}xwang8\}@sjtu.edu.cn, slu@cs.ucla.edu

Abstract—The problem of function computation in large scale heterogeneous wireless sensor networks (WSNs) is studied. Suppose $n$ sensors are placed in a disk network area with radius $\frac{n^\alpha}{\log n}$, where $\alpha$ is a positive constant. The sensors are located heterogeneously around the sink node, i.e., the density of sensors decreases as the distance from the sink node increases. At one instant, each sensor is assigned an input bit. The target of the sink is to compute a function $f$ of the input bits, where $f$ is either a symmetric or the identity function. Energy-efficient algorithms based on inhomogeneous tessellation of the network are designed and the corresponding optimal energy consumption scaling laws are derived. We show that the proposed algorithms are indeed optimal (except for some polylogarithmic terms) by deriving matching lower bounds on the energy consumption required to compute $f$. At last, based on the results obtained in this paper as well as those obtained by previous works, some discussions and comparisons are presented. We observe that, (i) the heterogeneity extent has a great impact on the computation of both symmetric function and identity function; (ii) the energy usage of computing symmetric function can be significantly smaller than that of computing identity function under certain parameter condition, i.e., performing in-network computation helps save energy.

Index Terms—Heterogeneity, in-network computation, symmetric function, identity function, wireless sensor networks (WSNs).

1 INTRODUCTION

Wireless sensor networks (WSNs) usually consist of numerous small sensor nodes which are capable of computing and communicating data. Mostly, the WSN has a sink node or a fusion center which is responsible for processing the sensor data and extracting the information it wants from the raw data. Often, the sink is not interested in all the raw data collected by those sensors. Rather, it only wants to know the value of a particular function of the raw data. For instance, in a fire alarm system, the sink may want to know the maximum value of the temperature data collected by the sensors. Traditional method is to gather all the raw data at the sink and then compute the desired function. However, this turns out to be very energy/time inefficient. Recently, \textit{in-network computation} is proposed to be a promising paradigm to reduce the overhead of WSNs \cite{1} \cite{2}. The basic idea of in-network computation is to perform computation in a distributed way, i.e., processing the data while transmitting them. This could lead to essential data compression and help save both energy and time consumption of WSNs. Lots of works have been done on in-network computation and we only focus on the most relevant ones which study in-network computation in a scaling law (asymptotic analysis) perspective.

The scaling law of wireless networks originates from the study of pure information delivery networks. In their seminal work \cite{3}, Gupta and Kumar investigate the capacity of uniform unicast networks and show that the per-node throughput capacity is $\Theta\left(\frac{1}{\sqrt{\frac{\log n}{n}}}\right)$. Following the framework of \cite{3}, researchers have studied the capacity of ad hoc wireless networks in different scenarios and aspects, including mobility \cite{4}, capacity-delay tradeoffs \cite{5}, multicast \cite{6} etc.

While the scaling law for pure information delivery networks has been extensively studied in the literature, asymptotic analysis for in-network computation is just emerging. Unlike pure information delivery networks whose goal is to deliver information without any processing, the aim of computation-oriented networks is to compute certain function (of the distributed data in the network) at the sink, which has the potential to reduce the communication overhead. Giridhar and Kumar \cite{8} take the first step to consider the impact of function computation on network throughput scaling. They study the computation of symmetric function, divisible function, type-sensitive and type-threshold function in both broadcast and multihop wireless networks and derive the maximum computing rates. Khude et al. \cite{9} study the computation of \textit{max} function in noiseless multihop wireless networks. Due to the fact that real channels are noisy, some propose to model the noise as binary symmetric channel with a certain crossover probability. For this, Ying et al. \cite{10} study the distributed computation of symmetric function in multihop noisy networks and demonstrate an algorithm which can compute any
symmetric function at the sink with an energy usage of \( \Theta \left( \left[ \log m \right] n \log \log n \left( \frac{\log n}{n} \right)^\gamma \right) \), where \( n \), \( m \), \( \gamma \) are the total number of nodes, the size of the measurement set and the path loss exponent respectively. Likewise, Li and Dai [11] consider the computation of divisible function and some special functions in multihop noisy networks and present energy efficient schemes. However, both [10] and [11] only give an achievable upper bound of the energy usage without offering matching lower bound, i.e., the optimality of the proposed algorithms are not ensured. In contrast, Karamchandani et al. [12] investigate the computation of both symmetric function and identity function in both noiseless and noisy grid networks. Matching lower bounds and upper bounds are provided for both energy and latency complexity, which verify the optimality of the computation protocols proposed in [12]. Beyond scaling law analysis, information theoretic methods (e.g., network coding) [13] [14], graph theoretic methods [15] as well as security issues [16] are also studied in the in-network computation literature.

Note that all the above mentioned works only consider in-network computation over homogeneous/uniform networks. However, many real-world wireless networks exhibit heterogeneity to some extent. For example, if a WSN is aimed at monitoring a volcano, as the distance to the volcano center increases, the sensor nodes density should decrease since the data near the volcano center is usually more critical. Indeed there have been a great deal of papers focusing on the scaling law analysis of pure information delivery heterogeneous networks [17] [18] [21] [22] [23], which indicate the great importance of heterogeneity. In this paper, we focus on the energy consumption scaling laws for the computation-oriented networks. Actually, lots of work on energy consumption scaling for in-network computation have been done in the literature [11] [10] [12] [9] since energy usage is a major concern in many practical networks, especially sensor networks. But, past computation protocols and energy complexity results obtained for homogeneous networks are not suitable for heterogeneous networks. Therefore, two fundamental questions arise:

- What is the impact of heterogeneity on energy consumption for in-network computation?
- Does in-network computation help reduce energy consumption when the network is heterogeneous?

To answer the above two questions quantitatively and to obtain a comprehensive understanding of the energy consumption for in-network computation over practical networks, we are motivated to study the energy consumption for function computation in heterogeneous networks in this paper. As in [12], we consider the computation of symmetric function and identity function, which are justified to be very useful and practical functions by previous works [8] [10] [20]. The notations of this paper are summarized in Table 1.

Our main contributions can be listed as follows:

- We identify three regimes, namely, slightly heterogeneous regime, significantly heterogeneous regime and heavily heterogeneous regime. We propose new network tessellation methods different from previous works on homogeneous networks for all the regimes. Based on the novel tessellation, we apply the intra-cell/inter-cell computation scheme [12] and thus build the framework for energy efficient computation in heterogeneous networks.
- For all the three heterogeneous regimes, we design and analyze energy efficient algorithms to compute symmetric function and identity function respectively. The optimality of the proposed algorithms is validated by deriving matching lower bounds on energy consumption.
- We observe that the heterogeneity extent can have a great impact on the minimum energy consumption for in-network computation. Specifically, for computation of symmetric function, as the network becomes more and more heterogeneous, the optimal energy usage will first keep unchanged (slightly heterogeneous regime), then increases (significantly heterogeneous regime) and finally decreases (heavily heterogeneous regime). For identity function, the relationship is a little bit complex and is reported in Subsection 2.6.

Due to the heterogeneity of the node distribution, compared with that of homogeneous network, the computation schemes become much more complicated and hard to analyze. Although our system model is basically the one-cluster network, it is straightforward to extend our analysis to the multi-cluster network. Throughout this paper, we are interested in large scale networks where the number of nodes tends to infinity. Note that our results apply to not only WSNs, but also any large scale wireless network, e.g., mobile internet, which desires to compute functions at the terminals. We believe that the results of this work could bring fundamental insights to designing large scale networks more energy efficiently. The remaining part of this paper is organized as
follows. In Section 2, we formulate the system model formally and present the main results of this work. In Section 3, we propose novel network tessellation methods based on which the algorithms for computing symmetric and identity function are designed. In Section 4, we prove the feasibility of the proposed algorithms. In Section 5, we analyze the energy complexity for the proposed algorithms. In Section 6, we derive the matching lower bounds for energy consumption of any feasible computation schemes. In Section 7, some discussions are provided, while in Section 8, we conclude this paper.

2 SYSTEM MODEL

2.1 Network Architecture

Consider a WSN of \( n \) sensor nodes located in a disk area \( O \). Each node is equipped with a radio transmitter and receiver and works in half-duplex mode, i.e., a node cannot transmit and receive signals at the same time. We assume that the network has already been synchronized so that the transmissions can be scheduled without conflicting. The radius of \( O \) is assumed to be \( n^{\alpha} \), where \( \alpha > 0 \) is a constant called network extension exponent. By this, we let the network area scale with the number of nodes. Note that this enables us to characterize networks with different density, i.e., dense networks (\( \alpha < \frac{1}{2} \)), extended networks (\( \alpha = \frac{1}{2} \)) and sparse networks (\( \alpha > \frac{1}{2} \)). Suppose that the sink node is located at the center of \( O \). Each of the \( n \) nodes chooses its location independently and randomly. Once a node has chosen its location, it remains static. To make the network heterogeneous, we let the nodes choose their positions heterogeneously, i.e., each node will have a large (low) chance to choose a position near to (distant from) the sink, as illustrated in Figure 1. Specifically, as in [17] [18], suppose the distance between a point and the sink to be \( \rho \), then the probability density function (PDF) for a node to choose this point as its position is \( h(\rho) \), which is specified as follows:

\[
h(\rho) = \frac{s(\rho)}{\int_{\Omega} s(||x||)dx},
\]

where \( s(\rho) = \min \{1, \rho^{-\delta} \} \) and the positive constant \( \delta \) is called the heterogeneity exponent. In Eqn.(1), \( x \) is the position vector with respect to the sink. The function \( s(\cdot) \) is used to model the network heterogeneity while the denominator in Eqn.(1) is for normalization of PDF. We remark that the heterogeneous model specified in (1) is widely adopted in the literature of scaling laws for heterogeneous wireless networks [17] [18] [21] [22] [23]. Although we assume that the sink happens to be the topology center of the heterogeneous network, the scaling results obtained in this paper also apply to the cases where the sink is not the center of the network. The reason is that we can always perform the final computation at the center (as if it is a sink) and then transmit the computation result from the center to the actual sink of the sensor network, which evidently does not change the energy scaling in this paper. In addition, we note that the model here is not only suitable for sensor networks but also suitable for the general heterogeneous wireless networks. Hence our analysis can cover lots of practical networks besides sensor networks. Now we identify three heterogeneous regimes according to the value of \( \delta \) as follows.

- **Slightly Heterogeneous Regime**, when \( 0 < \delta < 2 \);
- **Significantly Heterogeneous Regime**, when \( 2 < \delta < 2 + \frac{1}{\alpha} \);
- **Heavily Heterogeneous Regime**, when \( \delta \geq 2 + \frac{1}{\alpha} \).

The denominator in Eqn.(1) can be simplified as follows:

\[
\int_{\Omega} s(||x||)dx = \Theta \left( \int_{1}^{n^{\alpha}} \rho s(\rho) d\rho \right) = \begin{cases} \Theta(1), & 0 < \delta < 2, \\ \Theta(n^{\alpha(2-\delta)}), & \delta > 2. \end{cases}
\]

Thereby, in the slightly heterogeneous regime, we have \( h(\rho) = \Theta(s(\rho)n^{\alpha(\delta-2)}) \), while in the other two regimes we have \( h(\rho) = \Theta(s(\rho)) \).

2.2 Energy Consumption Model

Due to the heterogeneity of the distribution of the nodes, the distances to nearest neighbors are different for different nodes. Thereby, we naturally let different nodes to exploit different transmission ranges \( r \). The relationship between the energy consumption and the transmission range, which is widely exploited in the literature [10], is as follows. Suppose the transmission range of a transmitter to be \( r \), then it will consume the transmitter \( r^\gamma \) units of energy to transmit one bit successfully, where \( \gamma \geq 2 \) is a constant called path loss exponent. The fundamental difference in transmission ranges may make the algorithm and analysis more complicated than that of previous works on homogeneous networks: in homogeneous networks, the transmission ranges are all the same and to compute the total energy consumption, one only needs to count the total number of transmissions, while in heterogeneous networks, one needs to sum all the energy usage up (incurred by transmitters with different transmission ranges) to give the overall energy consumption. We note that the presented energy model does not take other factors such as sleep scheduling and duty-cycling into account. This

![Fig. 1. Illustration of Heterogeneous WSNs.](image-url)
work tries to capture the most essential feature of energy issues in wireless networks and gives a first glance at the energy consumption for function computation in heterogeneous networks.

2.3 Interference Model
As for the interference model, we exploit the traditional protocol model with different transmission ranges [3] in this work. Specifically, suppose that $X_i$ is a transmitter and $X_j$ is the intended receiver and $X_k$ is any other simultaneous transmitter. Let $r_i$ and $r_k$ be the transmission ranges of $X_i$ and $X_k$ respectively. Then the transmission from $X_i$ to $X_j$ is successful iff both of the following two conditions hold.

- $||X_i - X_j|| \leq r_i$
- $||X_j - X_k|| \geq (1+\Delta)r_k$, for any simultaneously active transmitter

where $\Delta$ is a positive constant called guard zone. This well-known protocol model, although being simple and omitting many details of a successful communication, still captures the most significant factor, interference, in wireless networks and is therefore widely adopted in the literature of asymptotic analysis. Actually, when considering the scaling law (complexity), it is a well-known result that the protocol model is equivalent to the more realistic physical model [3].

2.4 Data and Function Model
Like [12], assume that at one instant, each sensor node is assigned with one binary input data, i.e., either ‘0’ or ‘1’. Although the input data is constrained to be binary, we believe that the extension to multi-bit input is not difficult under our framework. The network is assumed to have been synchronized already.

The goal of the sink is to compute a particular function $f$ of the raw data. First, we consider the function $f$ to be symmetric function, i.e., the output of $f$ is independent of the order of the input data. Formally speaking, a binary function $f$ is called symmetric function iff for any input $y \in \{0,1\}^n$ and any permutation $\pi$ on $\{1,2,\ldots,n\}$:

$$f(y_1,y_2,\ldots,y_n) = f(y_{\pi(1)},y_{\pi(2)},\ldots,y_{\pi(n)})$$

In other words, the value of a symmetric function only depends on the histogram of the input and knowing the histogram function of the raw data is enough to compute any symmetric function. In binary case, a binary symmetric function only depends on the total number of ‘1’s among the input data. Note that many useful practical functions are symmetric function, e.g., histogram, average, sum, maximum, majority, parity. Therefore, symmetric function is a widely investigated function type in the literature of in-network computation [8] [10] [12].

Afterwards, we study the computation of the identity function. The output of the identity function is exactly the input data vector. Thus, computing the identity function is equivalent to gathering all the raw data at the sink and no in-network computation is allowed actually, which is a traditional information delivery procedure. Investigating the energy scaling of computing the identity function could serve as a benchmark: any function can be computed based on the results of the identity function, so the identity function is the hardest function to compute. By comparing the energy usage of computing symmetric function to that of computing identity function, one may see how much energy consumption reduction is obtained by performing in-network computation. Due to the special feature of it, the identity function is also widely studied [11] [12] [20].

2.5 Design Goal
The design goal of our algorithm is to compute the desired target function at the sink with the minimum total energy consumption, which is the most significant performance metric widely accepted in the literature of in-network computation [10] [11] [12]. We do not handle the fairness of the energy consumption among the wireless nodes. However, we note that in our heterogeneous model, to compute a symmetric function, any feasible scheme cannot reduce the energy usage of any single node by more than a poly-log factor compared with the proposed algorithm in this paper. Thus, even for each individual node, our proposed algorithm is nearly optimal. Throughout this paper, we denote the probability of event $E$ as $P(E)$ and say $E$ happens with high probability (w.h.p.) if $\lim_{n \to \infty} P(E) = 1$.

2.6 Main Results
The main results of this paper, i.e., the optimal energy complexity for computation of symmetric function and identity function, are summarized in Table 2. The notation $\Theta$ hides the poly-log terms and is adopted for better readability. The symbol $(\cdot)^+$ is defined as $x^+ = \max\{x,0\}$.

A graphical representation is presented in Figure 2. Sub-figure 2(a) plots the relationship between the energy complexity for computing symmetric function and the heterogeneity exponent $\delta$. Two turning points exist, which divides the heterogeneous networks into three regimes identified in Subsection 2.1. In the slightly heterogeneous regime, as $\delta$ increases, the energy complexity remains unchanged and is equal to that of homogeneous networks (but the optimal algorithm to achieve the complexity is very different). In the significantly heterogeneous regime, the energy complexity increases with $\delta$, while it decreases in the heavily heterogeneous regime. An intuitive analysis of this phenomenon is shown in Section 7.

Sub-figures 2(b)(c)(d) report the optimal energy complexity for computing identity function with three sub-figures plotting the energy complexity in three regions, i.e., $\gamma < \frac{3}{2\alpha+1}$, $\frac{3}{2\alpha+1} \leq \gamma < 3$ and $\gamma \geq 3$, respectively. The energy complexity of symmetric function is also depicted for comparison. We observe that: sometimes,
the energy for computing symmetric function is significantly smaller than that for computing identity function while sometimes they are the same (when the two curves overlap, they are simply represented by the curve of the single symmetric function, i.e., the green solid one). Hence, the heterogeneity exponent \( \delta \) has a great impact on energy consumption for both symmetric function and identity function.

We further observe that for both symmetric function and identity function, the energy scaling does not change with \( \delta \) in the slightly heterogeneous regime \((0 < \delta < 2)\), i.e., the heterogeneity does not influence the energy scaling in this regime. Specifically, when \( \delta = 0 \), the heterogeneous network degenerates into a homogeneous one and the optimal energy scaling should be \( \Theta \left( n^{\alpha \gamma - \frac{2}{\gamma} + 1} \right) \) and \( \tilde{\Theta} \left( n^{\alpha \gamma - \frac{2}{\gamma} + \frac{2}{\alpha}} \right) \) for symmetric function and identity function respectively. Actually, this coincides with the result in [12] for homogeneous networks after proper parameter transfer, which corroborates the result of this paper.

However, we remark that although the energy scaling in the slightly heterogeneous regime is the same as that of homogeneous networks, the optimal algorithms to achieve this optimal energy scaling are quite different as we will see later.

### 3.1 Slightly Heterogeneous Regime

In this subsection, we propose a tessellation method and corresponding computation scheme so as to compute symmetric function and identity function in slightly heterogeneous networks. We begin with the definition of an increasing sequence \( \{ S_k, 0 \leq k \leq m \} \) as follows, where \( m = \frac{n^2 - n^{\frac{\alpha (2 - \delta)}{2}}}{\log n} \),

\[
S_k = \left( n^\frac{\alpha}{2} - n^{\frac{\alpha (2 - \delta)}{2}} \frac{1}{\log n} + k \right)^{\frac{2}{\alpha}} n^{\alpha - \frac{1}{\gamma^2} (\log n)^{\frac{\gamma}{2}}}.
\]

Thus, we have \( S_0 = 1, S_m = n^\alpha \). Afterwards, we first tessellate the disk network into \((m + 1)\) annuluses centered...
at the sink. Define \( S_{-1} = 0 \). As illustrated in Figure 3, the inner and outer radius of the \( k \)-th radius are \( S_{k-1} \) and \( S_k \) respectively, where \( i = 0, 1, \ldots, m \). Thus, the distance between two adjacent annuluses \( x_k \) \((k = 1, 2, \ldots, m)\) can be calculated as \( x_k = S_k - S_{k-1} \).

Then, we further tessellate each circle into several circular segments. We call the circle with radius \( S_k \) the \( k \)-th circle. We tessellate the \( k \)-th circle into equal circular segment with length \( l_k = \frac{17\pi x_k}{\log n} \), as can be seen in Figure 3. The feasibility of this tessellation is proved in Section 4.

After that, we connect the endpoints of each circular line with the sink. The connection lines will divide the corresponding annulus into some small sectors, which we call cells. As verified in Section 4, each cell have \( \Theta(\log n) \) nodes w.h.p. and we could randomly pick one point in each cell as a cell center, which is illustrated in Figure 3. Thereby, we could invoke the traditional intra-cell/inter-cell in-network computation scheme here. The detailed procedure, consisting of three phases, is specified in the following three sub-sections. Note that we omit the discussion of those nodes within the unit circle centered at the sink since the node distribution is homogeneous there according to Eqn.(1). If the number of nodes inside the unit circle tends to be infinity, i.e., \( \alpha(2 - \delta) < 1 \), we can also construct a spanning tree and invoke the intra-cell/inter-cell multi-hop computation scheme as [12]. Otherwise, the number of nodes inside the unit circle is bounded (at most a poly-log complexity), i.e., \( \alpha(2 - \delta) \geq 1 \), and the nodes inside the unit circle just exploit one hop transmission to reach the sink. In what follows, we focus on areas outside the unit circle.

### 3.1.1 Constructing A Spanning Tree

The intra-cell/inter-cell protocol is tree-based, thereby we need to construct a spanning tree, where the cell centers are the tree nodes. Here, a simple construction method is: for an arbitrary cell center \( H_j \), find an adjacent cell \( C_i \) (by adjacent, we mean two cells that have common points) which is in the inner annulus, as illustrated in Figure 3. Then \( H_i \), the cell center of \( C_i \), is the parent node of \( H_j \). Performing this for all the cell centers (including the possible cell centers within the unit circle centered at the sink), a spanning tree is constructed.

### 3.1.2 Intra-cell Phase

In each cell, one node is randomly elected as the cell center. Denote \( H_j \) and \( n_j \) as the cell center and the number of nodes in the \( j \)-th cell, where \( n_j = \Theta(\log n) \). Then, each node takes turns to transmit its data to its corresponding cell center in one hop directly. Thus, each cell center collects all the data within its cell, including its own data. If the goal function \( f \) is a symmetric function, each cell center computes the total number of ‘1’s among the data of its cell and saves the computation result in its storage. Otherwise, if the goal function \( f \) is the identity function, each cell center skips this local computation process and just stores all the raw data within its cell. The intra-cell phase is ended.

### 3.1.3 Inter-cell Phase

In the inter-cell phase, the goal function is computed through a tree structure of the cell centers. Suppose the goal function \( f \) to be a symmetric function. Suppose the intra-cell computation result (the number of ‘1’s within its cell) stored in a cell center \( H_j \) is \( \eta_j \). The computation starts from the cell centers of those outer annuluses, i.e., the bottom of the routing tree, as illustrated in Figure 4. Each cell center, say \( H_j \), receives the sum computation result from its child nodes and then adds those sums together with its own intra-cell computation result \( \eta_j \) to get its new sum. Then it delivers this sum computation result to its parent node. A cell center performs the inter-cell computation only if all of its child nodes have already done it. When all the cell centers have finished the inter-cell computation, the sink node gets the total sum of \( \sum_j \eta_j \), i.e., the total number of ‘1’s in the whole network. Therefore, any symmetric function \( f \) can be calculated and the inter-cell phase is ended. Next, we turn to the case where the goal function \( f \) is the identity function. In this case, the cell centers perform no in-network computation and just deliver all the raw data to the sink in a multi-hop tree-based manner. The routing structure is the same to that of computing symmetric function (Figure 4). The difference is that rather than just transmit a computation result, the cell centers need to transmit all the associated raw data to its parent node. When the sink gets all the raw data of the entire network, the inter-cell phase is finished.

### 3.2 Heavily Heterogeneous Regime

As in the slightly heterogeneous regime, we first define an increasing sequence \( \{\tilde{S}_k, 0 \leq k \leq \tilde{m}\} \) as follows,

\[
\tilde{S}_k = \left( \frac{1}{\sqrt{n} - k} \right)^{\frac{1}{2}} n^{\frac{1}{4}},
\]

where \( \tilde{m} = \lceil \sqrt{n} - \log^2 n \rceil \). Thereby, we have \( \tilde{S}_0 = 1, \tilde{S}_{\tilde{m}} = n^{\frac{1}{4}} (\log n)^{-\frac{1}{2}} \). Define \( \tilde{R} = n^{\frac{1}{4}} \log n \). Denote \( T(a, b) \) as the annulus region centered at the sink with inner radius \( a \) and outer radius \( b \). Then, we partition the entire network area into four parts as follows:
• First part: \( T(0, 1) \). In this part, the network is homogeneous and the number of nodes inside this region is \( \Theta(n) \). We could invoke the multihop intra-cell/inter-cell computation protocol proposed by previous works on homogeneous works [12]. The details are omitted.

• Second part: \( T(1, S'_{m}) \). Unlike the first part, the node distribution in this part is heterogeneous. Based on the sequence \( \{S_k\} \), we perform a similar tessellation as in the slightly heterogeneous regime in the previous subsection. Specifically, we first tessellate this region into \( \tilde{m} \) annuli centered at the sink with radius \( \tilde{S}_k \) respectively as in Figure 3. We define the width of the \( k \)-th annulus as \( \tilde{x}_k = \tilde{S}_k - \tilde{S}_{k-1} \)

We further tessellate the \( k \)-th circle into some equal circular line segments with length \( \tilde{l}_k = c\tilde{x}_k \log n \), where \( c \) is some positive constant to be determined. Connecting the sink with the endpoints of those circular segments tessellates the \( k \)-th annulus into several sectors, which we call cells. Then an intra-cell/inter-cell computation scheme is used as in the slightly heterogeneous networks in Subsection 3.1. The feasibility of this scheme is proved in Section 4.

• Third part: \( T(\tilde{S}_{m'}, \tilde{R}) \). Due to the heterogeneity, the node distribution in this part is so sparse that it is very tough for a node to find a neighbor, i.e., the nearest node is also very far away. For nodes in this part, we let them just deliver their data to the sink in one hop directly. Since the total number of nodes in third part in only \( \Theta(1) \) (seeLemma 5), this will not hurt the total energy consumption. Any other feasible schemes cannot reduce the energy consumption of the nodes in this part further (except for poly-log terms) since, for these nodes, the distances to the nearest neighbor nodes are in the same order of the distances to the sink (except for poly-log terms).

• Fourth part: \( T(\tilde{R}, n^\alpha) \). We claim that there is no node in this part, w.h.p., which is guaranteed by Lemma 6.

3.3 Significantly Heterogeneous Regime

In the significantly heterogeneous regime, for areas outside the unit circle, we perform similar tessellation as in the heavily heterogeneous regime with the sequence \( \{S_k, 1 \leq k \leq \tilde{m}\} \) defined as follows:

\[
S_k = \left(\frac{1}{\sqrt{n-k}}\right)^{\frac{2}{\alpha(\alpha-2)}} n^{\frac{\alpha}{\alpha-2}}.
\]

Note that \( \tilde{m} = \left\lfloor n^{\frac{1}{2}} - n^{\frac{1}{2} - \frac{\alpha(\alpha-2)}{2}} \right\rfloor \), which is different from \( m \) defined in the heavily heterogeneous regime. Hence, \( S_0 = 1, \tilde{S}_{\tilde{m}} = n^\alpha \) and all the network area is tessellated. Other quantities are similarly defined as in heavily heterogeneous regime in Subsection 3.2: \( \tilde{x}_k = \tilde{S}_k - \tilde{S}_{k-1}, \tilde{l}_k = c\tilde{x}_k \log n \). In the significantly heterogeneous regime, we apply the similar intra-cell/inter-cell computation scheme. Note that, different from heavily heterogeneous regime, every node transmits data in a multihop fashion in the significantly heterogeneous regime because the node density is still relatively high even in the most distant regions.

4 Feasibility of the Proposed Algorithms

In this section, we prove the feasibility of the proposed algorithm in Section 3. The proof framework for the three regimes is analogous. Hence, we handle the slightly heterogeneous regime first and then extend the results to the other two regimes.

4.1 Slightly Heterogeneous Regime

In this subsection, we prove the feasibility of the proposed algorithm in slightly heterogeneous regime. The feasibility is composed of two components:

• The tessellation is feasible, i.e., the \( k \)-th circle is long enough to be divided into several circular segments of length \( l_k = \frac{17\pi S_k}{\log n} \). To see this, it is sufficient to show that \( S_k = \omega(l_k), \forall 1 \leq k \leq m \) since the circumference of the \( k \)-th circle is \( 2\pi S_k \). This is guaranteed by Lemma 1.

• The inter-cell phase is feasible, i.e., each cell will have at least one node. Otherwise, there exist some cells with no node and the inter-cell phase routing tree will break because of lack of cell center. For this regard, it is sufficient to prove that each cell has \( \Theta(\log n) \) nodes, w.h.p.. This is ensured by Lemma 2.

As long as the above two constraints are satisfied, the proposed algorithm is proved feasible. Now we give the related lemmas as follows.

Lemma 1: In the proposed algorithm for slightly heterogeneous networks, \( S_k = \omega(l_k) \).

Proof: Please refer to Section 1 in the supplementary file.

Lemma 2: In the proposed algorithm for slightly heterogeneous networks, each cell has \( \Theta(\log n) \) nodes within it, w.h.p..

Proof: Please refer to Section 2 in the supplementary file.

4.2 Significantly Heterogeneous Regime and Heavily Heterogeneous Regime

The proof of the feasibility of the proposed algorithm in the significantly heterogeneous regime is quite similar to and even simpler than (only needs Lemma 3 and Lemma 4) that in the heavily heterogeneous regime. So we only present the proof in the heavily heterogeneous regime.

Recall that we divide the computation protocols into four parts for the heavily heterogeneous regime in Subsection 3.2. The feasibility of the first part is tackled by
previous works on homogeneous networks [10] and is
omitted here. We first prove the feasibility of the second
part. The proof also consists of two components, i.e., the
feasibility of the network tessellation and the existence
of at least one node in each cell. The previous one is
verified in Lemma 3, while the latter one is validated in
Lemma 4.

Lemma 3: In the proposed algorithm for heavily het-
erogeneous networks, \( \tilde{S}_k = \omega(\tilde{l}_k) \).

Proof: The proof is similar to that of Lemma 1 and
is omitted here.

Lemma 4: In the proposed algorithm for heavily heter-
erogeneous networks, if we choose the constant in the
algorithm to be \( c > \frac{16\sqrt{2}\pi \delta}{(\gamma-2)(\min\{\frac{1}{\sqrt{2}}, 2^{\frac{1}{2\gamma}}-1\})^2} \), each cell
has \( \Theta(\log n) \) nodes within it w.h.p.

Proof: The proof is similar to that of Lemma 2 and
is omitted here.

Afterwards, we endeavor to validate the feasibility of
the third part, i.e., the number of nodes in the third part
is quite small, which is guaranteed by Lemma 5.

Lemma 5: In heavily heterogeneous networks, the total
number of nodes within the third part is at most \( \widetilde{O}(1) \)
w.h.p.

Proof: Please refer to Section 3 in the supplementary
file.

Finally, we verify the feasibility of the fourth part,
i.e., there is no node within the fourth part with high
probability. This is finished by Lemma 6.

Lemma 6: There is no node within the fourth part in
the heavily heterogeneous regime w.h.p.

Proof: Please refer to Section 4 in the supplementary
file.

5 ENERGY COMPLEXITY OF THE PROPOSED
ALGORITHMS

In this section, we analyze the energy consumption com-
plexity for computing symmetric function and identity
function of the presented algorithm. We will first focus
on the slightly heterogeneous regime and then present
the results of the other two regimes.

5.1 Slightly Heterogeneous Regime

In this subsection, we derive the energy scaling for
computing symmetric function and identity function in
slightly heterogeneous networks respectively.

5.1.1 Symmetric Function

In this sub-subsection, we analyze the total energy usage
of the proposed algorithm for computing symmetric
function in the slightly heterogeneous regime. We re-
mark that in order to compute a symmetric function,
the network does not need to deliver all the raw data.
Rather, the network only needs to transmit the histogram
(number of ‘1’ s in the binary case) to the sink. Because
there are \( n \) sensor nodes in the network in total, trans-
mitting the number of ‘1’ s needs at most \( \lceil \log_2 n \rceil = \widetilde{O}(1) \)
bits. Since we ignore the poly-log terms when deriving
the energy scaling, this is equivalent to transmitting one
single bit. The energy consumption scaling for symmet-
ic function is concluded in the following theorem.

Theorem 1: In slightly heterogeneous networks, the
proposed algorithm can compute any symmetric func-
tion with an energy consumption no more than
\( \widetilde{O}\left(n^{\alpha\gamma - \frac{1}{2} + \frac{3}{2}}\right) \) w.h.p.

Proof: Please refer to Section 5 in the supplementary
file.

5.1.2 Identity Function

In this sub-subsection, we analyze the total energy us-
age of the proposed algorithm for computing identity
function in slightly heterogeneous networks. The result
is concluded as the following theorem.

Theorem 2: In slightly heterogeneous networks, the
proposed algorithm can compute the identity function
with an energy consumption no more than \( \widetilde{O}\left(n^{\alpha\gamma - \frac{1}{2} + \frac{1}{2}}\right) \)

w.h.p.

Proof: Please refer to Section 6 in the supplementary
file.

5.2 Heavily Heterogeneous Regime and Signifi-
cantly Heterogeneous Regime

In this section, we present the energy consumption scal-
ing in heavily heterogeneous regime and significantly
heterogeneous regime.

Theorem 3: In heavily heterogeneous networks, the
proposed algorithm can compute any symmetric func-
tion with an energy consumption no more than \( \widetilde{O}\left(n^{\frac{\alpha\gamma}{2}}\right) \)

w.h.p.

Proof: Please refer to Section 7 in the supplementary
file.

Theorem 4: In heavily heterogeneous networks, the
proposed algorithm can compute the identity function
with an energy consumption no more than \( \widetilde{O}\left(n^{\max\{\frac{\alpha\gamma}{2}, 1\}}\right) \) w.h.p.

Proof: Please refer to Section 8 in the supplementary
file.
Theorem 5: In significantly heterogeneous networks, the proposed algorithm can compute any symmetric function with an energy consumption no more than $\Theta\left(\frac{n^{\alpha\gamma}}{\gamma^2} - \alpha\delta + 1 - \frac{2}{3} + 2\alpha\right)$ w.h.p..

Proof: The proof is similar to that of Theorem 3 and is omitted here.

Theorem 6: In significantly heterogeneous networks, the proposed algorithm can compute the identity function with an energy consumption no more than $\Theta\left(\frac{3\gamma}{3-\gamma} \left(\frac{n^{\alpha\gamma}}{\gamma^2} - \frac{2}{3} + 3\alpha\right)^{\frac{1}{3}}\right)$ w.h.p..

Proof: The proof is similar to that of Theorem 4 and is omitted here.

6 MATCHING LOWER BOUNDS OF THE ENERGY COMPLEXITY

In this section, we derive matching lower bounds on the energy complexity in order to verify the optimality of the proposed algorithm for computing symmetric function and identity function. The proof framework for all the three heterogeneous regimes is about the same. Hence, we only present the proof of slightly heterogeneous regime as an example in the following.

6.1 Symmetric Function

In this subsection, we derive the lower bound of energy usage for computing symmetric function in slightly heterogeneous networks.

Consider an arbitrary cell A (the cell with shadow in Figure 5). Define those cells that have common points with A as A’s neighbor cells (those with stars in Figure 5). Define four adjacent cells of the neighbor cells of A as the sub-neighbor cells (those with hexagons in Figure 5). Denote cell A, its neighbor cells and sub-neighbor cells together as area B. Supposing that cell A is within the $k$-th annulus, then we have the following key lemma.

Lemma 7: To compute the histogram (the number of ‘1’s in binary case) function, a kind of symmetric function, the nodes inside the area B will consume at least $\Theta\left(x^\gamma_k\right)$ units of energy.

Proof: Please refer to Section 9 in the supplementary file.

Note that such area B consists of $\Theta(1)$ cells, which means that we can select such cell A with $\Theta(1)$ percent of all the cells and avoid any overlapping between different area B’s. Thus, the total energy consumption is at least $\Theta\left(\sum_{k=1}^{m} x^\gamma_k \times \frac{2\alpha}{3}\right) = \Theta\left(n^{\alpha\gamma-\frac{2}{3}+1}\right)$, which has already been derived in sub-subsection 5.1.1. Thereby, the optimality of the proposed algorithm in slightly heterogeneous regime is verified. By performing similar analysis, one can prove the optimality of the algorithm for symmetric function computation in the other two heterogeneous regimes.

6.2 Identity Function

The derivation of the matching lower bound for energy consumption scaling of the identity function is presented in the supplementary file due to space limitation.

7 DISCUSSION

In this section, we offer some intuitive explanation on the main results obtained in this work, which is summarized in Figure 2 and Table 2.

As presented in Sub-figure 2(a), the energy complexity of computing symmetric function has two turning points, which are determined by the two intersection points of the three heterogeneous regimes. In slightly heterogeneous regime, the energy complexity keeps unchanged and is equal to the homogeneous regime ($\delta = 0$). Hence, in this regime, heterogeneity does not influence the energy consumption basically. We note that although the energy usage for homogeneous networks and slightly heterogeneous networks are the same (omit poly-log terms), the optimal algorithms are quite different. To achieve the minimum energy usage, one should apply the proposed algorithm in this paper, which is based on a heterogeneous network tessellation. On the opposite, to achieve the minimum total energy consumption in homogeneous networks, a uniform tessellation based computation protocol is sufficient as in [10] [12]. When $\delta$ becomes larger, the network goes into significantly heterogeneous regime and it is quite interesting that the energy complexity increases with $\delta$. The essential reason is that in the distant (far from the sink) regions of the network, the node density becomes quite low. The distribution there is so sparse that it is hard for nodes there to find neighbor nodes, i.e., the distance to the nearest node is still large. According to our energy model, large transmission range leads to large energy overhead. Hence, the total energy consumption increases with $\delta$. But, in sharp contrast, as $\delta$ keeps increasing, the network goes into heavily heterogeneous regime and the energy consumption decreases with $\delta$. The basic reason is that in heavily heterogeneous regime, the network becomes so heterogeneous that the effective network area actually shrinks. Equivalently speaking, in those distant network...
areas, the node distribution is so sparse that there is no node at all with high probability (Lemma 6)! Hence, the network nodes only occupy part of the network area indeed. Due to this shrinking effect, the distance between nodes (communication range) decreases and thus the total energy consumption declines.

The expression and turning points of energy usage for computing the identity function is much more complicated (as illustrated in Sub-figures 2(b)(c)(d)) and is therefore tough to give an intuitive analysis. However, we still notice that the energy usage for the identity function still keeps unchanged in slightly heterogeneous regime, which is a common feature of symmetric function.

Remember that computing symmetric function is indeed an in-network computation while computing identity function is equivalent to gathering all the raw data, i.e., no in-network computation is allowed actually. From the comparison of energy complexity for symmetric function and identity function, as reported in Sub-figures 2(b)(c)(d), performing in-network computation sometimes help save energy usage while sometimes not (the energy usage of symmetric function and identity function can sometimes be equal). Thus, the analytical results obtained in this work could guide practical engineers to design energy-efficient heterogeneous networks by taking the advantage of in-network computation more precisely.

8 Conclusion

The energy complexity of function computation in heterogeneous wireless networks is studied in this paper. The target function is either a symmetric function or the identity function, which are both widely accepted very useful and practical functions. We investigate the impact of heterogeneity on the energy consumption in wireless networks and find that the heterogeneity extent greatly influences the minimum total energy usage as well as the optimal scheme for computing functions. We observe that performing in-network computation sometimes reduce energy usage while sometimes not. Our theoretical results offer fundamental insights into how to take the advantage of in-network computation to design more energy-efficient large scale wireless networks, e.g., WSNs and large scale mobile Internet, etc.

9 Acknowledgment

This paper is supported by National Fundamental Research Grant (No. 2011CB302701); NSF China (No. 61272129, 61221001, 61025016); Shanghai Basic Research Key Project (No. 11JC1405100); Shanghai International Cooperation Project (No. 13510711300); China Ministry of Education Fok Ying Tung Fund (No. 122002); China Ministry of Education New Century Excellent Talent (No. NCET-10-0580); National Key project (No. 2012ZX03001009).

References


**Xuanyu Cao** received the B.E. degree in electronic engineering from Shanghai Jiaotong University, Shanghai, China, in 2013. He is now pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering at University of Maryland, College Park, MD. His research interests are in the area of scaling law analysis as well as signal processing in wireless networks.

**Xinbing Wang** received the B.S. degree (with hons.) from the Department of Automation, Shanghai Jiaotong University, Shanghai, China, in 1998, and the M.S. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2001. He received the Ph.D. degree, major in the Department of Electrical and Computer Engineering, minor in the Department of Mathematics, North Carolina State University, Raleigh, in 2006. Currently, he is a professor in the Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai, China. Dr. Wang has been an associate editor for IEEE/ACM Transactions on Networking and IEEE Transactions on Mobile Computing, and the member of the Technical Program Committees of several conferences including ACM MobiCom 2012, ACM MobiHoc 2012, 2013, IEEE INFOCOM 2009-2014.

**Songwu Lu** received both his MS and PhD degrees in Electrical Engineering from UIUC. He is currently a professor at UCLA Computer Science Department. His research interests include wireless networking, mobile systems, cloud computing, and network security.