Aggregation Capacity of Wireless Sensor Networks: Extended Network Case

Cheng Wang, Changjun Jiang, Yunhao Liu, *Senior Member*, *IEEE*,
Xiang-Yang Li, *Senior Member*, *IEEE*, and Shaojie Tang

Abstract—A critical function of wireless sensor networks (WSNs) is data gathering. One is often only interested in collecting a specific function of the sensor measurements at a sink node, rather than downloading all the raw data from all the sensors. In this paper, we study the capacity of computing and transporting the specific functions of sensor measurements to the sink node, called *aggregation capacity*, for WSNs. We focus on random WSNs that can be classified into two types: *random extended WSN* and *random dense WSN*. All existing results about aggregation capacity are studied for *dense WSNs*, including random cases and arbitrary cases, under the *protocol model* (ProM) or *physical model* (PhyM). In this paper, we propose the first aggregation capacity scaling laws for *random extended WSNs*. We point out that unlike random dense WSNs, for random extended WSNs, the assumption made in ProM and PhyM that each successful transmission can sustain a constant rate is over-optimistic and unpractical due to *transmit power limitation*. We derive the first result on aggregation capacity for random extended WSNs under the *generalized physical model*. Particularly, we prove that, for the *type-sensitive divisible perfectly compressible functions* and *type-threshold divisible perfectly compressible functions*, the aggregation capacities for random extended WSNs with n nodes are of order $\Theta((\log n)^{-\frac{\alpha}{2}-1})$ and $\Theta(\frac{(\log n)^{-\alpha/2}}{\log\log n})$, respectively, where $\alpha > 2$ denotes the power attenuation exponent in the generalized physical model. Furthermore, we improve the aggregation throughput for general divisible perfectly compressible functions to $\Omega((\log n)^{-\frac{\alpha}{2}})$ by choosing $\Theta(\log n)$ sensors from a small region (relative to the whole region) as sink nodes.

Index Terms—Wireless sensor networks, data aggregation, aggregation capacity

1 Introduction

Wireless sensor networks (WSNs) are composed of nodes with the capabilities of sensing, communication and computation. One important application of wireless sensor networks (WSNs) is data gathering, i.e., sensor nodes transmit data, possibly in a multi-hop fashion, to a sink node. Actually, one is often only interested in collecting a relevant function of the sensor measurements at a sink node, rather than downloading all the data from all the sensors. Hence, it is necessary to define the capacity of computing and transporting specific functions of sensor measurements to the sink node. Since *in-network aggregation* plays a key role in improving such capacity for WSNs, we can reasonably call such capacity *aggregation capacity* for WSNs.

In this paper, we focus on scaling laws of the aggregation capacity for WSNs. Gupta and Kumar [1] initiated the study of capacity scaling laws for large-scale ad hoc wireless networks. The main advantage of studying scaling laws is to highlight qualitative and architectural properties of the system without getting bogged down by too many details [1], [2]. Generally, the capacity scaling laws of a network are directly determined by the adopted network models, including *deployment models*, *scaling models* and *communication models*, besides the pattern of traffic sessions. According to the controllability of a network, Gupta and Kumar [1] defined two types of deployment models: *arbitrary networks* and *random networks*. In terms of scaling methods, there are two types of scaling network models, i.e., *dense networks* and *extended networks*. Moreover, the *protocol model* (ProM), *physical model* (PhyM) and *generalized physical model* (GphyM, also called *Gaussian Channel model*, [3]) are three typical communication models. Following these models, most works focus on the capacities for different traffic sessions, such as *unicast*, *broadcast*, *multicast*, *anycast*, and *many-to-one* session, etc. Data aggregation of WSNs studied in this paper can be regarded as a special case of many-to-one sessions. The involvement of *in-network aggregation* [4] makes it more complex than the general data collecting in many-to-one sessions. Naturally, aggregation capacity scaling laws have characteristics different from the capacity of any other sessions, which is worth studying.

There exists some literature dealing with scaling laws of the aggregation capacity for different functions, e.g., [4]–[8]. To the best of our knowledge, almost all related works, for both random networks and arbitrary networks, only have considered the dense network model, and the results are all derived under *fixed-rate communication model* [9], including ProM and PhyM [1]. Hence, in this work, we study aggregation capacity scaling laws for the random extended WSN, contrary to existing theoretical results that apply only to dense WSNs.

- *C. Wang and C. Jiang are with the Department of Computer Science and Engineering, Tongji University, Shanghai, China, and with the Key Laboratory of Embedded System and Service Computing, Ministry of Education, China. E-mail: 3chengwang@gmail.com, cjjiang@tongji.edu.cn.*
- *Y. Liu is with Tsinghua National Laboratory for Information Science and Technology (TNLIST), School of Software, Tsinghua University, Beijing, China, and with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, China. E-mail: yunhaoliu@gmail.com.*
- *X.-Y. Li and S. Tang are with the Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA.*
E-mail: xli@cs.iit.edu, tangshaojie@gmail.com.

Manuscript received 11 Dec. 2011; revised 31 July 2012; accepted 09 Sep. 2012. Date of publication 17 Sep. 2012; date of current version 09 June 2014.
Recommended for acceptance by M. Eltoweissy.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TC.2012.230

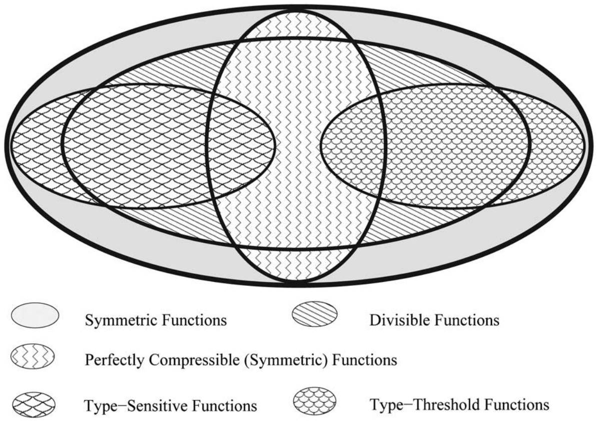


Fig. 1. Aggregation functions of interest.

Since the basic assumption in ProM and PhyM, i.e., any successful transmission can sustain a constant rate, is indeed over-optimistic and unpractical in extended networks, we use generalized physical model to capture the nature of wireless channels better.

We design an original aggregation scheme comprised of the tree-based routing and TDMA transmission scheduling. This scheme hierarchically consists of *local aggregation phase* and *backbone aggregation phase*. Based on this original aggregation scheme, we adopt two techniques to improve the aggregation capacity. The first one is to introduce *block coding* strategy into the local aggregation phase, by which the throughput during the local aggregation phase is improved; the second one is to devise the *parallel transmission scheduling*, which can improve the throughputs during both the local aggregation phase and backbone aggregation phase except for the bottleneck on the sink node.

Main Contributions: We now summarize major contributions of this paper as follows:

1. For a special subclass of symmetric functions, called *divisible functions*, we design an aggregation scheme, denoted by $\mathcal{A}_{N,n}$, for the random extended WSN (RE-WSN), and derive the general result on the achievable aggregation throughput, depending on the characteristics of specific aggregation functions. Please see the relations among functions of interest studied in this paper in Fig. 1. (**Theorem 1**)
2. For a special subclass of divisible functions, called *divisible-perfectly-compressible aggregation functions* (DPC-AFs), we show that under the scheme $\mathcal{A}_{N,n}$, the achievable aggregation throughput for RE-WSN is of order $\Omega((\log n)^{-\frac{\alpha}{2}-1})$. (**Theorem 2**)
3. For a special subclass of DPC-AFs, called *type-threshold* DPC-AFs, such as *max* (or *min*), *range*, and various kinds of *indicator functions*, we devise a new aggregation scheme, denoted by $\mathcal{A}^{bc}_{N,n}$, by integrating the *block coding* [4] into the scheme $\mathcal{A}_{N,n}$. We show that under $\mathcal{A}^{bc}_{N,n}$ the achievable aggregation capacity for RE-WSN is of order $\Omega(\frac{(\log n)^{-\alpha/2}}{\log\log n})$. (**Theorem 3**)
4. For two subclasses of DPC-AFs, i.e., *type-sensitive* DPC-AFs (e.g., *average* function) and *type-threshold* DPC-AFs, we derive the upper bounds on aggregation capacities, and prove that our schemes $\mathcal{A}_{N,n}$ and $\mathcal{A}^{bc}_{N,n}$ are optimal for type-sensitive DPC-AFs and type-threshold DPC-AFs, respectively. Combining the lower bounds (Theorem 2 and Theorem 3) with the upper bounds (**Theorem 4** and **Theorem 5**), we obtain the tight bounds on aggregation capacities for type-sensitive DPC-AFs and type-threshold DPC-AFs are of order $\Theta\Big((\log n)^{-\frac{\alpha}{2}-1}\Big)$ and $\Theta\Big(\frac{(\log n)^{-\alpha/2}}{\log\log n}\Big)$, respectively. (**Theorem 6**)
5. By choosing $\Theta(\log n)$ sensors randomly from a small region, of which the area is an infinitesimal proportion, i.e., $\Theta(\frac{\log n}{n})$, of the total area of the RE-WSN, as the sink nodes, we design a parallel aggregation scheme, denoted by $\mathcal{A}^{p}_{N,n}$. Under this scheme, we introduce a technique called *parallel transmission scheduling*, and show that for DPC-AFs, the measurements from all n sensors can be aggregated into those $\Theta(\log n)$ sink nodes at a throughput of order $\Omega((\log n)^{-\frac{\alpha}{2}})$. (**Theorem 7**)

The rest of the paper is organized as follows. In Section 2, we introduce the system model. In Section 3, we propose the specific aggregation schemes for RE-WSNs to derive the achievable aggregation capacity. In Section 4, we compute the upper bounds on the aggregation capacities for type-sensitive divisible perfectly compressible functions and divisible type-threshold perfectly compressible functions, then obtain the tight capacity bounds for two types of functions. In Section 5, we make efforts in improving further the aggregation capacity for RE-WSNs by introducing the technique called parallel transmission scheduling. In Section 6, we review the related work. In Section 7, we draw some conclusions and discuss the future work.

2 System Model

Throughout the paper, we mainly consider the events that happen with high probability (*w.h.p.*) as the scale of a network (the number of sensors in a network) goes to infinity.

2.1 Aggregation Capacity

We consider a random WSN, denoted by $\mathcal{N}(\mathfrak{a}^2, n)$,[1] where n sensors are placed uniformly at random in a square $\mathcal{A}(\mathfrak{a}^2) = [0, \mathfrak{a}] \times [0, \mathfrak{a}]$, and a sensor, denoted by s_0, is chosen as the sink node. Like in most models considered in related works, every sensor node $s_i, 0 \leq i \leq n-1$, periodically generates measurements of the environment that belong to a fixed finite set $\mathcal{M}$ with $|\mathcal{M}| = \mathfrak{m}$, and the function of interest is then required to be computed repeatedly. Intuitively, the capacity for WSNs depends on the aggregation functions of interest to the sink node, [4], [8].

2.1.1 Formal Notations

Define the aggregation function of interest to the sink node as $\mathbf{g}_n : \mathcal{M}^n \rightarrow \mathcal{G}_n$; furthermore, for any k, $1 \leq k \leq n$, define the function of the sensor measurements as $\mathbf{g}_k : \mathcal{M}^k \rightarrow \mathcal{G}_k$, where $\mathcal{G}_k$ is the range of $\mathbf{g}_k$. Suppose that each sensor has an associated block of N readings, known a *priori* [4]. We define N rounds of measurements from all n sensors as a *processed unit*. From a practical perspective, only the same round of measurements, which are usually attached to the same time

[1] The results in this paper also apply to the random network where the sensors are placed in the region $\mathcal{A}(\mathfrak{a}^2) = [0, \mathfrak{a}] \times [0, \mathfrak{a}]$ according to a Poisson point process of density $\lambda = \frac{n}{\mathfrak{a}^2}$.

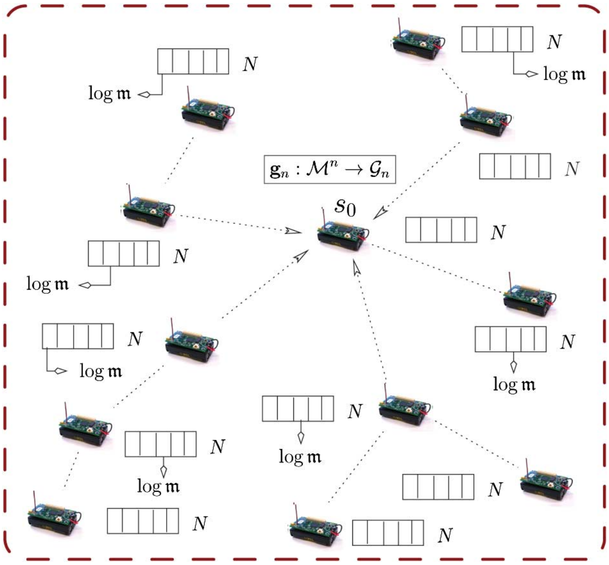


Fig. 2. System model. An *aggregation unit* consists of N measurements. Each measurement can be indicated by $\log \mathfrak{m}$ bits, [10].

stamps, are requested and permitted to be aggregated. Please see the illustration in Fig. 2. We first introduce some notations.

- A processed unit consisting of N rounds of measurements from all n sensors is denoted by a $n \times N$ matrix $M^{n\times N} \in \mathcal{M}^{n\times N}$, where $M^{n\times N}(i,j)$ is the jth measurement of sensor node s_i, $M^{n\times N}(i,\cdot)$ is the ith row of $M^{n\times N}$, i.e., the block of N measurements of sensor node s_i, and $M^{n\times N}(\cdot,j)$ is the jth column of $\mathcal{M}^{n\times N}$, i.e., the set of the jth measurements of all n sensor nodes.
- For a k-vector $M^k = [M_1, M_2, \ldots, M_k]^T \in \mathcal{M}^k$, where $M_i \in \mathcal{M}$, define $\mathbf{g}_k(M^k) := \mathbf{g}_k(M_1, M_2, \ldots, M_k)$.
- Given a matrix $M^{k\times N}$, $1 \le k \le n$, define
- $\mathbf{g}_k^{\mathrm{N}}(M^{k\times N}) := (\mathbf{g}_k(M^{k\times N}(\cdot,1)), \ldots, \mathbf{g}_k(M^{k\times N}(\cdot,N)))$.
- An aggregation scheme dealing with the aggregation of N rounds of measurements, denoted by $\mathcal{A}_{N,n}$, determines a sequence of message passings between sensors and computations at sensors. Under the scheme $\mathcal{A}_{N,n}$, input any $M^{n\times N} \in \mathcal{M}^{n\times N}$, output a result $\mathbf{g}_n^{\mathrm{N}}(M^{n\times N})$ at the sink node s_0.

2.1.2 Capacity Definition

First, we give the definition of *achievable aggregation throughput* for WSNs. All the logs in this paper are to the base 2.

Definition 1. *A throughput of $\Lambda(n)$ bits/s is achievable for a given aggregation function $\mathbf{g}_n$ if there is an aggregation scheme, denoted by $\mathcal{A}_{N,n}$, by which any $M^{n\times N} \in \mathcal{M}^{n\times N}$ can be aggregated into $\mathbf{g}_n^{\mathrm{N}}(M^{n\times N})$ at the sink node within $T(\mathcal{A}_{N,n})$ seconds, where $\Lambda(n) \le \frac{N\cdot\log\mathfrak{m}}{T(\mathcal{A}_{N,n})}$, $\mathfrak{m} = |\mathcal{M}|$, and $N\log\mathfrak{m}$ is the total number of bits representing N measurements from each sensor.*

Based on Definition 1, we define the *aggregation capacity* for random WSNs.

Definition 2. *For a given aggregation function $\mathbf{g}_n$, we say that the aggregation capacity of a class of random WSNs is of order $\Theta(f(n))$ bits/s for $\mathbf{g}_n$, if there are constants $c > 0$ and $c < d < +\infty$ such that*

$$\lim_{n\to+\infty} \Pr(\Lambda(n) = c\cdot f(n) \text{ is } \textit{achievable} \text{ for } \mathbf{g}_n) = 1,$$

$$\liminf_{n\to+\infty} \Pr(\Lambda(n) = d\cdot f(n) \text{ is } \textit{achievable} \text{ for } \mathbf{g}_n) < 1.$$

2.1.3 Aggregation Functions of Interest

We focus our attention to the *symmetric functions*, which are invariant with respect to permutations of their arguments. That is, for $1 \le k \le n$, and for all permutation σ, it holds that $\mathbf{g}_k(M_1, M_2, \ldots, M_k) = \mathbf{g}_k(\sigma(M_1, M_2, \ldots, M_k))$. From an application standpoint, many natural functions of interest, including most statistical functions, belong to this class. Symmetric functions embody the *data centric* paradigm [4], [11], where it is the data generated by a sensor that is of primary importance, rather than its identity [4]. Furthermore, we limit the scope of this work to a special subclass of symmetric functions, called *divisible functions* [4], which can be computed in a *divide-and-conquer* fashion. Divisible functions are usually deemed as the general functions in the study of data aggregation in WSNs.

Next, we introduce other two special subclasses of symmetric functions, called *type-sensitive functions* and *type-threshold functions*, respectively.

Type-Sensitive Functions: A symmetric function $\mathbf{g}_n(\cdot)$ is said to be *type-sensitive* if there exist a constant θ with $0 < \theta < 1$, and an integer n_0, such that for $n \ge n_0$, and any $i \le n - \lceil \theta n \rceil$, given any subset $\{M_1, M_2, \ldots, M_i\}$, there are two subsets of values $\{M'_{i+1}, M'_{i+2}, \ldots, M'_n\}$ and $\{M''_{i+1}, M''_{i+2}, \ldots, M''_n\}$, such that

$$\mathbf{g}_n(M_1, \ldots, M_i, M'_{i+1}, \ldots, M'_n) \ne \mathbf{g}_n(M_1, \ldots, M_i, M''_{i+1}, \ldots, M''_n).$$

For a n-vector $M^n \in \mathcal{M}^n$, the *mode* of M^n (i.e., the value that occurs most frequently), the *mean* of M^n, the *median* of M^n, and the *standard deviation* of M^n all belong to the type-sensitive functions.

Type-Threshold Functions: A symmetric function $\mathbf{g}_n(\cdot)$ is said to be *type-threshold* if there exists a nonnegative $\mathfrak{m}$-vector ξ, called the *threshold vector*, such that

$$\mathbf{g}_n(M^n) = \mathbf{g}'_{\mathfrak{m}}(\phi(M^n)) = \mathbf{g}'_{\mathfrak{m}}(\min\{\phi(M^n), \xi\}),$$

for all $M^n \in \mathcal{M}^n$, with min signifying element-wise minimum, where $\phi(M^n)$ is called *type-vector* that is defined as

$$\phi(M^n) = [\phi_1(M^n), \phi_2(M^n), \ldots, \phi_{\mathfrak{m}}(M^n)],$$

with $\phi_k(M^n) := |\{s_i : M_i = k\}|$.

For a n-vector $M^n \in \mathcal{M}^n$, the *max* function, the *min* function and the *range* function $(\max_i M_i - \min_i M_i)$ of M^n, the *kth largest value* of M^n, the *mean of the k largest values* of M^n, and the *indicator* function $I\{M_i = k, \text{ for some } i\}$ all belong to type-threshold functions.

Note that there are indeed some symmetric functions which are neither type-sensitive nor type-threshold, [4]. Please see the illustration in Fig. 1.

Specially, we focus on an important class of symmetric functions called *divisible-perfectly-compressible* aggregation functions (DPC-AFs). A divisible function is *perfectly compressible* if *all information concerning the same measurement round contained in two or more messages can be perfectly aggregated in a single new packet of equal size (in order sense)*, [8]. The following lemma is straightforward.

Lemma 1. *For any divisible-perfectly-compressible aggregation function (DPC-AF) $\mathbf{g}_k$, $1 \le k \le n$, it holds that $|\mathcal{G}_k| = \Theta(\mathfrak{m})$, where $\mathcal{G}_k$ is the range of the function $\mathbf{g}_k$.*

We mainly consider two subclasses of DPC-AFs, i.e., *type-sensitive* DPC-AFs and *type-threshold* DPC-AFs. A DPC-AF is *type-sensitive* (or *type-threshold*) if it is a *type-sensitive function* (or *type-threshold function*). Please see the illustration in Fig. 1.

Intuitively, the value of a type-sensitive function cannot be determined if a large enough fraction of the arguments are unknown, whereas the value of a type-threshold function can be determined by a fixed number of known arguments. A representative case of type-sensitive DPC-AFs is the *average function*; while, the typical type-threshold DPC-AFs include *max* (or *min*), *range*, and various kinds of *indicator functions*.

2.2 Communication Model

A communication model can be defined as a *interference-safe feasible family* [12] in which each element is a set consisting of the links that can transmit simultaneously without negative effects, or in order sense, on each other in terms of link rate. Generally, there are two types of communication models in the research of capacity bounds: *adaptive-rate communication model* and *fixed-rate communication model*.

2.2.1 Adaptive-Rate Communication Model (ACM)

Under the *adaptive-rate communication model*, the reliably transmission rate is determined based on a continuous function of the receiver's SINR (Signal to Interference plus Noise Ratio). Generally, any communication pair $v_i \rightarrow v_j$ can establish a direct link, over a channel of bandwidth B, of rate $R(v_i, v_j) = B\log(1 + \frac{1}{\eta} \cdot \mathrm{SINR}(v_j))$. When $\eta > 1$, the receiver can achieve the maximum rate that meets a given BER requirement under a specific modulation and coding scheme. When $\eta = 1$, the adaptive-rate channel model is also called *generalized physical model* (GphyM) [3], [13]. It is practically assumed that all nodes are individually power-constrained under GphyM, i.e., for any node v_i, it transmits at a constant power $P_i \in [P_{min}, P_{max}]$, where P_{min} and P_{max} are some positive constants. The receiver v_j receives the signal from the transmitter v_i with strength $P_i \cdot \ell(v_i, v_j)$, where $\ell(v_i, v_j)$ indicates the path loss between v_i and v_j. Any two nodes can establish a direct communication link, over a channel of bandwidth B, of rate

$$R(v_i, v_j) = B\log\left(1 + \frac{P_i \cdot \ell(v_i, v_j)}{N_0 + \sum_{v_k \in A(i)} P_k \cdot \ell(v_k, v_j)}\right), \quad (1)$$

where $N_0 > 0$ is the ambient noise power at the receiver, and $A(i)$ is the set of nodes transmitting concurrently with v_i. The wireless propagation channel typically includes path loss with distance, shadowing and fading effects. As in [3], [9], [13], [14], we assume that the channel gain depends only on the Euclidean distance between a transmitter and receiver, and ignore shadowing and fading.

2.2.2 Fixed-Rate Communication Model (FCM)

To simplify the analysis of the system, Gupta and Kumar [1] defined the *fixed-rate communication model* as the abstraction of the wireless communication model, under which if the value of a defined conditional expression is beyond some threshold, the transmitter can send successfully to the receiver at a specific constant data rate; otherwise, it can not send any, i.e., the transmission rate is assumed to be a binary function. The *protocol model* (ProM) and *physical model* (PhyM) defined in [1] both belong to the fixed-rate channel model. The former's conditional expression is the fraction of the distances from the intended transmitter and other ones to a specific receiver; the latter's conditional expression is SINR. Obviously, the validity of FCM is based on the following assumption.

Assumption 1. *Any successful transmission can sustain the rate of a fixed constant order.*

2.3 Network Scaling Model

We clarify the differences between the *random extended WSN* (RE-WSN) and *random dense WSN* (RD-WSN).

2.3.1 Criteria of Scaling Patterns

In the research of network capacity scaling laws, there are two typical models in terms of scaling patterns of the network: *extended scaling model* and *dense scaling model* [2], [13], [15]. The major difference between the engineering implications of these two scaling models is related to the classical notions of *interference-limitedness* and *coverage-limitedness*. The *dense networks* tend to have dense deployments so that signals are received at the users with sufficient signal-to-noise ratio (SNR) but the throughput is limited by interference among the simultaneous transmissions. That is, all nodes can communicate with each other with sufficient SNR, and the throughput can only be *interference-limited*. While, the *extended networks* tend to have sparse deployments so that the throughput is mainly limited by the ability to transmit signals to the users with sufficient SNR. That is, the source and destination pairs are at increasing distance from each other, so both interference limitation and power limitation can come into play.

Recall that a given random network $\mathcal{N}(\mathfrak{a}^2, n)$ is constructed by placing uniformly at random n sensors in a square deployment region $\mathcal{A}(\mathfrak{a}^2) = [0, \mathfrak{a}] \times [0, \mathfrak{a}]$. Next, we examine the scaling characteristics of $\mathcal{N}(\mathfrak{a}^2, n)$ as $n \rightarrow \infty$, according to the relation between $\mathfrak{a}$ and n.

First, we recall some existing results about random Euclidean Minimum Spanning Tree (EMST) from [16]–[21]. Let $\mathcal{T}(\mathfrak{a}^2, n)$ denote the length of the longest edge of EMST built on the set of n nodes in the random network $\mathcal{N}(\mathfrak{a}^2, n)$. For the random variable $\mathcal{T}(\mathfrak{a}^2, n)$, according to a related result proven in [16], Li et al. [20], [21] proposed the following lemma.

Lemma 2. ([20], [21]). *For the random variable $\mathcal{T}(\mathfrak{a}^2, n)$, and for any real number $\nu(n)$, it holds that*

$$\lim_{n\rightarrow\infty} \Pr\left(n\pi \cdot \left(\frac{\mathcal{T}(\mathfrak{a}^2, n)}{\mathfrak{a}}\right)^2 - \ln n \leq \nu(n)\right) = \frac{1}{e^{e^{-\nu(n)}}},$$

where $\ln$ *is the natural logarithm to the base e.*

Based on Lemma 2, let $\nu(n) = -\ln\ln n$, we get that $\mathcal{T}(\mathfrak{a}^2, n) = \Omega(\mathfrak{a} \cdot \sqrt{\frac{\ln n}{n}})$, i.e., $\Omega(\mathfrak{a} \cdot \sqrt{\frac{\log n}{n}})$ with high probability, e.g., $1 - \frac{1}{n}$; and let $\nu(n) = \ln n$, we obtain that $\mathcal{T}(\mathfrak{a}^2, n) = O(\mathfrak{a} \cdot \sqrt{\frac{\log n}{n}})$ with high probability, e.g., at least $1 - \frac{1}{n}$. Hence, we have

$$\lim_{n\rightarrow\infty} \Pr\left(\mathcal{T}(\mathfrak{a}^2, n) = \Theta\left(\mathfrak{a} \cdot \sqrt{\frac{\log n}{n}}\right)\right) \geq 1 - \frac{1}{n}. \quad (2)$$

Now, we can define the criterion of the *extended scaling* versus *dense scaling* networks according to the order of the length of $\mathcal{T}(\mathfrak{a}^2, n)$, i.e., $\Theta(\mathfrak{a} \cdot \sqrt{\frac{\log n}{n}})$.

Definition 3. *Given a random network $\mathcal{N}(\mathfrak{a}^2, n)$, it is* dense scaling *if $\mathfrak{a} \cdot \sqrt{\frac{\log n}{n}} = O(1)$, i.e., $\mathfrak{a} = O(\sqrt{\frac{n}{\log n}})$, with high probability; it is* extended scaling, *if $\mathfrak{a} \cdot \sqrt{\frac{\log n}{n}} = \omega(1)$, i.e., $\mathfrak{a} = \omega(\sqrt{\frac{n}{\log n}})$, with high probability.*

2.3.2 RE-WSN vs. RD-WSN

The *extended network* and *dense network* are the representative cases of the extended and dense scaling models, respectively. They are specialized into the cases of $\mathfrak{a} = \sqrt{n}$ and $\mathfrak{a} = 1$, respectively, i.e., they can be denoted by $\mathcal{N}(n, n)$ and $\mathcal{N}(1, n)$, respectively. A random dense WSN (RD-WSN) represents the scenario where the monitoring region is fixed, and the scale of a network is expanding as the density of sensors is increasing; while, a random extended WSN (RE-WSN) represents the scenario where the density of sensors is fixed, and the scale of the network is expanding as the area of monitoring region is increasing.

Denote the sets of all sensors in the RE-WSN and RD-WSN by $\mathcal{S}_e$ and $\mathcal{S}_d$, respectively. Furthermore, denote $\mathcal{S}_e = \mathcal{S}_d := \{s_0\} \cup \{s_1, s_2, \ldots, s_{n-1}\}$, where s_0 is the sink node and s_i $(1 \leq i \leq n-1)$ denotes the sensor node.

2.3.3 Communication Models in Scaling Models

Now, we analyze the combinations of communication models and scaling models, and make a choice of communication model for this paper. Following the setting in [13], the channel power gain is given by $\ell(v_i, v_j) = \min\{1, d_{ij}^{-\alpha}\}$ in the extended scaling network; and it is given by $\ell(v_i, v_j) = d_{ij}^{-\alpha}$ in the dense scaling network. Here, $d_{ij} = \|v_i v_j\|$ is the Euclidean distance between two nodes v_i and v_j, $\alpha > 2$ denotes the power attenuation exponent [13].

- **FCM in Dense Scaling Networks:** Gupta and Kumar [1] only defined the FCM, including protocol model and physical model, in dense networks under which Assumption 1 is convincing because the large enough SINR (generally of order $\Theta(1)$) can be obtained. Thus, most results of the aggregation capacity [4], [5], [22], [23] derived under FCM are reasonable for dense networks.
- **GphyM in Dense Scaling Networks:** In dense networks, FCM can be regarded as a perfect abstraction of the generalized physical model (GphyM). Indeed, the capacity derived under GphyM can be equally derived by using FCM, and vice versa.
- **FCM in Extended Scaling Networks:** In extended networks, according to Definition 3, under any routing scheme for a random network $\mathcal{N}(\mathfrak{a}^2, n)$, there must be, *w.h.p.*, a link of distance of order $\Omega(\mathfrak{a} \cdot \sqrt{\frac{\log n}{n}})$, i.e., $\omega(1)$. By Equation (1), the SINR of such a link is too small to contribute to a constant rate. In other words, Assumption 1 is over-optimistic for random extended networks.
- **GphyM in Extended Scaling Networks:** The GphyM can appropriately embody the continuous link rate in extended networks, which is the reason why most existing studies on the capacity for extended networks are implemented under GphyM, [3], [9], [13], [24].

3 Lower Bounds on Aggregation Capacity

To simplify the description, we define a notion called *network lattice* that is frequently used in the design of aggregation schemes and the analysis of network characteristics.

Definition 4. (Network Lattice). *For a network $\mathcal{N}(\mathfrak{a}^2, n)$, divide the deployment region $\mathcal{A}(\mathfrak{a}^2)$ into a lattice consisting of subsquares (cells) of side length $\mathfrak{l}$, we call the generated lattice network lattice, and denote it by $\mathbb{L}(\mathfrak{a}, \mathfrak{l}, n)$.*

From now on, we focus on the RE-WSN $\mathcal{N}(n, n)$.

3.1 Aggregation Scheme for General Divisible Functions

Our aggregation scheme, denoted by $\mathcal{A}_{N,n}$, is designed based on the network lattice $\mathbb{L}_1 = \mathbb{L}(\sqrt{n}, 2\sqrt{\log n}, n)$. To simplify the description, we ignore the details about the integers, and assume that the number of rows (or columns) $\frac{\sqrt{n}}{2\sqrt{\log n}}$ is always an integer, which has no impact on the results due to the characteristics of scaling laws issue. Taking the cell in top left corner as the origin with a 2-dimensional index (0,0), we give each cell in $\mathbb{L}_1$ an index in the order from left to right and from top to bottom, i.e., the index of the cell in bottom right corner is (δ, δ), where $\delta = \delta(n) = \frac{\sqrt{n}}{2\sqrt{\log n}} - 1$. By using VC Theorem (Theorem 25 in [3]), we have the following result,

Lemma 3. *For all subsquares of side length $\mathfrak{l} = 2\sqrt{\log n}$ in the deployment region $\mathcal{A}(n)$, the number of sensors in those cells is uniform w.h.p., within $\frac{\log n}{2} < n_{i,j} < 8 \log n$.*

The proof of Lemma 3 is very similar to that of Lemma 18 in [20] (based on VC theorem [25]). Note that the involved constants in Lemma 3, i.e., $\frac{1}{2}$ and 8, do not change the final scaling laws of aggregation capacity indeed.

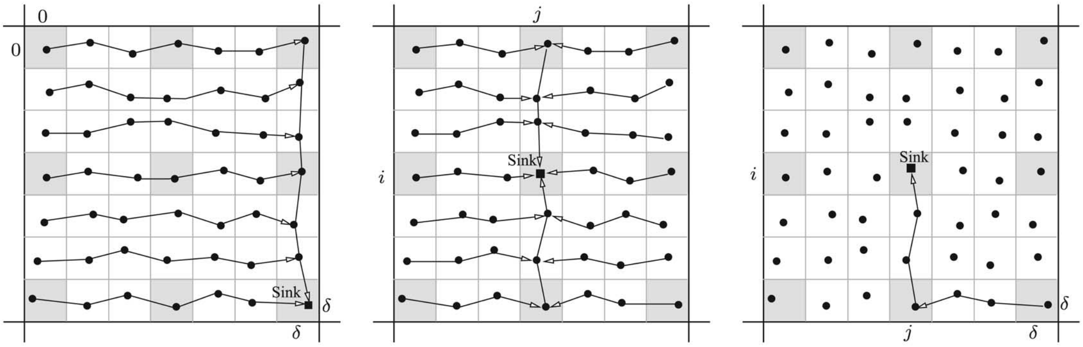
3.1.1 Aggregation Routing Scheme

The aggregation routing tree is divided into two levels, i.e., the *aggregation backbones* and *local aggregation links*.

Aggregation Backbones: In the network lattice $\mathbb{L}_1$, from the cells, except for that one containing s_0, we randomly choose one sensor from each cell, and obtain a set, denoted by $\mathcal{B}_s$ consisting of $(\delta + 1)^2 - 1$ nodes (sensors). Then, define the set $\mathcal{B} := \mathcal{B}_s \cup \{s_0\}$ as *backbone set*. We call the nodes in $\mathcal{B}$ as *aggregation stations*, or simply as *stations*.

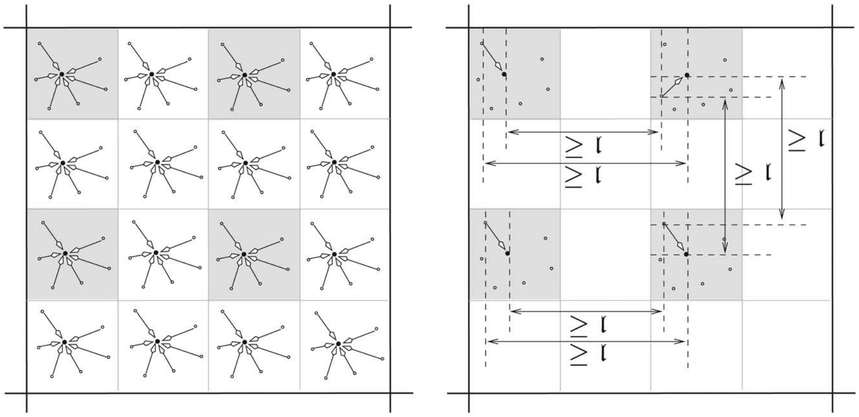
We assume that the sink s_0 is located in the cell $C_{\delta,\delta}$. By connecting the adjacent aggregation stations in the same rows, as illustrated in Fig. 3(a), we construct the *horizontal backbones* of the aggregation routing; by connecting the adjacent stations in δth column, we build the *vertical backbone*. For the general case in terms of the location of sink s_0, we introduce the extending method as follows: Assume that s_0 is located in the cell $C_{i,j}$, the difference in the construction of routing backbone is that the vertical backbone is built in jth column, instead of in δth column, as illustrated in Fig. 3(b). In fact, we can build a multihop path between the station in cell $C_{\delta,\delta}$ and the sink in $C_{i,j}$, as illustrated in Fig. 3(c). It can be proven that any such path is definitely not the bottleneck throughout routing. In words, the location of s_0 does not change scaling laws of the aggregation capacity.

Local Aggregation Links: In each cell of $\mathbb{L}_1$, all sensors, except for the station, communicate with the station in a single hop. Please see the illustration in Fig. 4(a).



(a) s_0 in $C_{\delta,\delta}$ (b) sink in $C_{i,j}$ (c) additional path

Fig. 3. Construction of aggregation backbones. (a) The case that we mainly focus on in this paper. (b) The general case in terms of the location of s_0. (c) The additional path from our final station to the sink s_0.



(a) Local Aggregation (b) Limiting Interference

Fig. 4. Local aggregation. (a) A 4-TDMA scheme is adopted. Each slot is further divided into $8\log n$ subslots that are assigned to all links in the cell. (b) 4-TDMA scheme guarantees that the distance between any receiver and the nearest unintended transmitters is of at least $\mathfrak{l} = 2\sqrt{\log n}$.

3.1.2 Aggregation Scheduling Scheme

In a global perspective, the aggregation scheduling scheme is divided into two phases: *local aggregation scheduling* and *backbone aggregation scheduling*. In both phases, all scheduled senders transmit with a same fixed power $P \in [P_{min}, P_{max}]$. In first phase, each aggregation station collects the measurements from the sensors in its assigned cell in $\mathbb{L}_1$. In second phase, the data in the aggregation stations are collected into the sink node (not simply level by level). Recall that we make a group of N rounds of measurements from all n sensors as a *processed unit*, denoted by a matrix $M^{n\times N}$ (Please refer to Section 2.1.1).

Local Aggregation Scheduling: Firstly, we use a 4-TDMA scheme to schedule the cells in $\mathbb{L}_1$, as illustrated in Fig. 4(a). In this phase, only the links completely contained in some cells are scheduled. From Lemma 3, the number of all links in each cell is *w.h.p.*, not more than $8\log n$. Then, we can further divide each slot of the 4-TDMA scheme into $8\log n$ subslots, by which we can ensure that all links can be scheduled once during a scheduling period that consists of $32\log n$ subslots.

Backbone Aggregation Scheduling: In this phase, the aggregated data are sent to the sink in a *pipelined fashion* [4], [8], and are aggregated on the way in each aggregation station. The backbone aggregation scheduling consists of two phases: *horizontal backbone phase* and *vertical backbone phase*. First, the data are horizontally aggregated into the stations in δth column; then the data are vertically aggregated into the sink node in the cell $C_{\delta,\delta}$.

In the initial state of horizontal scheduling, for all i and j, the aggregation station in cell $C_{i,j}$, denoted by $b_{i,j}$, holds N aggregation functions values of N rounds of measurements from all sensors in $C_{i,j}$ that can be denoted by a matrix $M^{n_{i,j}\times N}$. Denote those N aggregation functions by

$$\mathbf{g}^{\mathrm{N}}_{n_{i,j}}(M^{n_{i,j}\times N}) = (\mathbf{g}_{n_{i,j}}(M^{n_{i,j}\times N}(\cdot, z)), z = 1, 2, \ldots, N).$$

By this time, denote all N rounds of data held by all $(\delta+1)^2$ stations as a matrix $M_h^{(\delta+1)^2\times N}$.

During the horizontal backbone phase, denote the set of $b_{i,j}$ and all its descendants by $D^h_{i,j}$, thus the cardinality of $D^h_{i,j}$ is $|D^h_{i,j}| = \Theta((j+1)\cdot\log n)$. Then, the aggregation function value of the kth round of data at station $b_{i,j}$ is denoted by

$$\mathfrak{b}^h_{i,j}(k) := \mathbf{g}_{|D^h_{i,j}|}(M_h^{|D^h_{i,j}|\times N}(\cdot, k)). \tag{3}$$

Here, $\mathfrak{b}^h_{i,0}(k) = \mathbf{g}_{n_{i,0}}(M^{n_{i,0}\times N}(\cdot, k))$, for $k = 1, 2, \ldots, N$.

In the initial state of the vertical backbone phase, all stations $b_{i,\delta}$ hold N aggregation function values of N rounds of data from the stations $b_{i,j}$, $0 \le j \le \delta$, i.e., $\mathfrak{b}^h_{i,\delta}(k)$, $1 \le k \le N$.

By this time, denote N rounds of data holden by all $\delta+1$ stations as a matrix $M_v^{(\delta+1)\times N}$. Denote the set of $b_{i,\delta}$ and all its descendants by $D^v_{i,\delta}$, then the cardinality of $D^v_{i,\delta}$ is $|D^v_{i,\delta}| = \Theta((i+1)\cdot\sqrt{n\log n})$. During the vertical backbone phase, the value of the aggregation function of the kth round of data at station $b_{i,\delta}$ is denoted by

$$\mathfrak{b}^v_{i,\delta}(k) := \mathbf{g}_{|D^v_{i,\delta}|}(M_v^{|D^v_{i,\delta}|\times N}(\cdot,k)). \tag{4}$$

Here, $\mathfrak{b}^v_{0,\delta}(k) = \mathfrak{b}^h_{0,\delta}(k)$, for $k = 1,2,\ldots,N$.

We adopt a 9-TDMA scheme to schedule the horizontal backbones, as illustrated in Fig. 3(a), and adopt a 3-TDMA scheme to schedule the vertical backbone. We design Algorithm 1 and Algorithm 2 to schedule the horizontal and vertical backbone aggregations in a *pipelined fashion*, respectively. Implementing two algorithms once, we can compute N aggregation function values of N rounds of measurements at the sink node. Before presenting these two algorithms, we define two sequences of sets:

For $h = 0,1,2$ and $v = 0,1,2$, define

$$\mathcal{H}_{h,v} := \{b_{i,j} | i \bmod 3 = h, \text{ and } j \bmod 3 = v\};$$

for $h = 0,1,2$, define $\mathcal{V}_{h,\delta} := \{b_{i,\delta} | i \bmod 3 = h\}$.

Algorithm 1. Horizontal Backbone Aggregation

```
Input: $\mathbf{g}^{\mathrm{N}}_{n_{i,j}}(M^{n_{i,j}\times N})$ at all stations, i.e., $M_h^{(\delta+1)^2\times N}$.
Output: $\mathfrak{b}^h_{i,\delta}(k)$ at all station $b_{i,\delta}$.
for $k = 1,2,\ldots,N,N+1,\ldots,N+\delta(n)-3$ do
    $k \to k'$;
    if $k > N$ then $N \to k$;
    else for $h = 0,1,2$ do
        for $v = 0,1,2$ do
            for $r = 1,\ldots,k$ do
                All $b_{i,j} \in \mathcal{H}_{h,v}$ are permitted to transmit; if it holds that $1 \le j \le \delta-1$, and
                1. $b_{i,j}$, $j \ge 1$, has already received $\mathfrak{b}^h_{i,j-1}(r)$ from $b_{i,j-1}$, and
                2. $b_{i,j+1}$ has not received $\mathfrak{b}^h_{i,j}(r)$ from $b_{i,j}$, then $b_{i,j}$ sends $\mathfrak{b}^h_{i,j}(r)$ to $b_{i,j+1}$; else if $j = 0$, and $b_{i,1}$ has not received $\mathfrak{b}^h_{i,0}(r)$, i.e., $\mathbf{g}_{n_{i,0}}(M^{n_{i,0}\times N}(\cdot,r))$ from $b_{i,0}$, then $b_{i,0}$ sends $\mathfrak{b}^h_{i,0}(r)$ to $b_{i,1}$.
        $k' \to k$
```

3.1.3 Aggregation Capacity Analysis

Aggregation capacity depends on the type of functions of interest. We propose a general method in the analysis of aggregation throughput, although we mainly focus on the divisible-perfectly-compressible functions (Section 2.1.3). Due to the hierarchical structure of our scheme, we carry out the analysis phase by phase.

Local Aggregation Phase: In this phase, since it is guaranteed that each link is scheduled at least once out of $32\log n$ time slots, Lemma 4 intuitively holds.

Lemma 4. *In the local aggregation phase, if each scheduled link can achieve the rate of $R_1(n)$ bits/s, then each link can sustain an average rate of $\Lambda_1(n) = \frac{R_1(n)}{32\log n}$ bits/s. Thus, it takes at most*

$$T_1(n) = \frac{N\cdot\log\mathfrak{m}}{\Lambda_1(n)} = \frac{32N\cdot\log\mathfrak{m}\cdot\log n}{R_1(n)} \tag{5}$$

seconds to finish the aggregation of N rounds of measurements from n sensors, i.e., a processed unit, at $(\delta+1)^2$ stations.

During the local aggregation phase, when block coding [4] is not adopted, since all data to be transmitted are the original measurements instead of the aggregated data, the throughput in this phase is independent of the type of aggregation functions. Next, we commence deriving the rate $R_1(n)$.

Lemma 5. *During the local aggregation phase, every scheduled link can achieve the rate of order*

$$R_1(n) = \Omega((\log n)^{-\frac{\alpha}{2}}). \tag{6}$$

Proof. Consider a given active cell, say $C_{i,j}$, in a time slot under the 4-TDMA scheme. Please see illustration in Fig. 4(b). First, we find an upper bound for the interference at the receiver (station). We notice that the transmitters in the eight closest cells are located at Euclidean distance at least $\mathfrak{l} = 2\sqrt{\log n}$ from the receiver (station) in $C_{i,j}$. The transmitters in the 16 next closest cells are at Euclidean distance at least $3\mathfrak{l}$, and so on. By extending the sum of the interferences to the whole plane, this is bounded as follows:

$$I_1(n) \le \sum_{k=1}^{\infty} P\cdot\frac{8k}{((2k-1)\cdot\mathfrak{l})^{\alpha}} \le \frac{P}{2^{\alpha-3}}\cdot(\log n)^{-\frac{\alpha}{2}}\cdot\sum_{k=1}^{\infty}\frac{k}{(2k-1)^{\alpha}}.$$

From $\alpha > 2$, $\sum_{k=1}^{\infty}\frac{k}{(2k-1)^{\alpha}}$ converges to a constant. Then, $I_1(n) = O((\log n)^{-\frac{\alpha}{2}})$.

Next, we find a lower bound on the strength of signal received from the transmitter. Since all links are limited within the same cells, the link length is at most $\sqrt{2}\cdot\mathfrak{l}$. Thus, the signal at the receiver can be bounded by

$$S_1(n) \ge P\cdot(\sqrt{2}\cdot\mathfrak{l})^{-\alpha} = P\cdot 2^{-\frac{3}{2}\alpha}\cdot(\log n)^{-\frac{\alpha}{2}}.$$

Hence, $S_1(n) = \Omega((\log n)^{-\frac{\alpha}{2}})$.

Finally, combining the fact that $I_1(n) = O((\log n)^{-\frac{\alpha}{2}})$ and $S_1(n) = \Omega((\log n)^{-\frac{\alpha}{2}})$, we obtain a lower bound on the rate of scheduled link as

$$R_1(n) = B\log\left(1+\frac{S_1(n)}{N_0+I_1(n)}\right) = \Omega\left((\log n)^{-\frac{\alpha}{2}}\right),$$

which completes the proof. □

Hence, according to Lemma 4 and Lemma 5, we have the following result,

Lemma 6. *When the technique of block coding is not used, the time cost of the local aggregation for N rounds of measurements is of order*

$$T_1(n) = O(N\cdot(\log n)^{1+\frac{\alpha}{2}}). \tag{7}$$

Backbone Aggregation Phase: First, we consider the horizontal backbone phase.

Lemma 7. *In the horizontal backbone phase, if each scheduled link can achieve the rate of $R_2^h(n)$ bits/s, then all horizontal backbones can sustain a rate of $\Lambda_2^h(n) = R_2^h(n) \cdot \frac{N}{9(N+\delta(n)-3)}$.*

Proof. According to Algorithm 1, each horizontal backbone can be scheduled in order at least N times out of $3 \times 3 \times (N + \delta(n) - 3)$ time slots. Thus, the lemma holds. □

Now, we start to derive the link rate $R_2^h(n)$.

Lemma 8. *During the horizontal backbone phase, every scheduled link can achieve the rate of order*

$$R_2^h(n) = \Omega((\log n)^{-\frac{\alpha}{2}}). \tag{8}$$

Proof. This lemma can be proven in a similar procedure to that of Lemma 5. □

Lemma 9. *To finish the horizontal backbone aggregation for N rounds of measurements, it takes at most*

$$T_2^h(n) = 9 \cdot \zeta_{max}^h \cdot (\log n)^{\frac{\alpha}{2}} \cdot (N + \delta(n) - 3) \tag{9}$$

seconds, where

$$\zeta_{max}^h = \max\{\zeta_{i,j}^h | 0 \le i \le \delta - 1; 0 \le j \le \delta - 1\},$$

with $\zeta_{i,j}^h = \log|\mathcal{G}_{|D_{i,j}^h|}|$, and $\mathcal{G}_{|D_{i,j}^h|}$ is the range of $\mathbf{g}_{|D_{i,j}^h|}$.

Proof. In this phase, the aggregation function value of the kth round of data at station $b_{i,j}$ is

$$\mathfrak{b}_{i,j}^h(k) := \mathbf{g}_{|D_{i,j}^h|}(M_h^{|D_{i,j}^h| \times N}(\cdot, k)).$$

Since the technique of block coding is not adopted here, the load of station $b_{i,j}$, denoted by $\Gamma_{i,j}^h$, is

$$\Gamma_{i,j}^h = \sum\nolimits_{k=1}^N \log|\mathcal{G}_{|D_{i,j}^h|}| = N \cdot \zeta_{i,j}^h.$$

Hence, in the horizontal backbone phase, the time cost of the aggregation for N rounds of measurements is at most

$$T_2^h(n) = \frac{N \cdot \zeta_{max}^h}{\Lambda_2^h(n)} = 9\zeta_{max}^h \cdot (\log n)^{\frac{\alpha}{2}} \cdot (N + \delta(n) - 3),$$

which completes the proof. □

Next, we analyze the vertical backbone phase in a similar method to the horizontal one. For concision, we omit some similar proofs.

Lemma 10. *In vertical backbone aggregation phase, the rate of each scheduled link is of order $R_2^v(n) = \Omega((\log n)^{-\frac{\alpha}{2}})$, and the vertical backbone can sustain a rate of order*

$$\Lambda_2^v(n) = \Omega\left((\log n)^{-\frac{\alpha}{2}} \cdot \frac{N}{3(N + \delta(n) - 3)}\right).$$

Proof. In this phase, a 3-TDMA scheme is adopted. By using the similar approach to Lemma 5 and Lemma 8, we can prove that $R_2^v(n) = \Omega((\log n)^{-\frac{\alpha}{2}})$. In Algorithm 2, the vertical backbone can be scheduled at least N times out of $3 \times (N + \delta(n) - 3)$ time slots, which proves the lemma. □

Lemma 11. *To finish the vertical backbone aggregation for N rounds of measurements, it takes at most*

$$T_2^v(n) = 3 \cdot \zeta_{max}^v \cdot (\log n)^{\frac{\alpha}{2}} \cdot (N + \delta(n) - 3) \tag{10}$$

seconds, where

$$\zeta_{max}^v = \max\{\zeta_{i,\delta}^v | 0 \le i \le \delta - 1\},$$

with $\zeta_{i,\delta}^v = \log|G_{|D_{i,\delta}^v|}|$, and $G_{|D_{i,\delta}^v|}$ is the range of $\mathbf{g}_{|D_{i,\delta}^v|}$.

Proof. In this phase, the aggregation function value of the kth round of data at station $b_{i,\delta}$ is

$$\mathfrak{b}_{i,\delta}^v(k) := \mathbf{g}_{|D_{i,\delta}^v|}(M_v^{|D_{i,\delta}^v| \times N}(\cdot, k)).$$

Since block coding is not used here, the load of station $b_{i,\delta}$, denoted by $\Gamma_{i,\delta}^v$, is $\Gamma_{i,\delta}^v = N \cdot \zeta_{i,\delta}^v$. Hence, to finish the vertical backbone aggregation for N rounds of measurements, it takes

$$T_2^v(n) = \frac{N \cdot \zeta_{max}^v}{\Lambda_2^v(n)} = 3\zeta_{max}^v \cdot (\log n)^{\frac{\alpha}{2}} \cdot (N + \delta(n) - 3),$$

which completes the proof. □

Algorithm 2. Vertical Backbone Aggregation

```
Input: $\mathfrak{b}_{i,\delta}^h(k)$ at all station $b_{i,\delta}$.
Output: $\mathbf{g}_n^{\mathrm{N}}(M^{n \times N})$ at the sink node $s_0$.
for $k = 1, 2, \ldots, N, N+1, \ldots, N + \delta(n) - 3$ do
    $k \rightarrow k'$;
    if $k > N$ then $N \rightarrow k$;
    else for $h = 0, 1, 2$ do
        for $r = 1, \cdots, k$ do
            All $b_{i,\delta} \in \mathcal{V}_{h,\delta}$ are permitted to transmit;
            if it holds that $1 \le i \le \delta - 1$, and
            (1) $b_{i,\delta}$, $i \ge 1$, has already received $\mathfrak{b}_{i-1,\delta}^v(r)$ from $b_{i-1,\delta}$, and
            (2) $b_{i+1,\delta}$ has not received $\mathfrak{b}_{i,\delta}^v(r)$ from $b_{i,\delta}$,
            then $b_{i,\delta}$ sends $\mathfrak{b}_{i,\delta}^v(r)$ to $b_{i+1,\delta}$;
            else if $i = 0$, and $b_{1,\delta}$ has not received
            $\mathfrak{b}_{0,\delta}^v(r)$, i.e., $\mathfrak{b}_{0,\delta}^h(r)$, from $b_{0,\delta}$,
            then $b_{0,\delta}$ sends $\mathfrak{b}_{0,\delta}^v(r)$ to $b_{1,\delta}$.
        $k' \rightarrow k$
```

According to Definition 1, we obtain Theorem 1.

Theorem 1. *The aggregation throughput under the scheme $\mathcal{A}_{N,n}$ with $N = \Omega(\frac{\sqrt{n}}{\sqrt{\log n}})$ is of order*

$$\Lambda(n) = \Omega\left(\frac{(\log n)^{-\frac{\alpha}{2}}}{\log n + \zeta^h_{max} + \zeta^v_{max}}\right), \tag{11}$$

where ζ^h_{max} and ζ^v_{max} are defined in Lemma 9 and Lemma 11, respectively.

Proof. First, we consider the total time cost, say $T(\mathcal{A}_{N,n})$, during which the aggregation functions of N rounds of measurements from n sensors are computed at the sink node. It holds that $T(\mathcal{A}_{N,n}) = T_1(n) + T^h_2(n) + T^v_2(n)$, and the aggregation throughput under the scheme $\mathcal{A}_{N,n}$ is of order

$$\Lambda(n) = \frac{N \log \mathfrak{m}}{T_1(n) + T^h_2(n) + T^v_2(n)}. \tag{12}$$

Based on Lemma 9 and Lemma 11, for $N = \Omega(\frac{\sqrt{n}}{\sqrt{\log n}})$, i.e., $N = \Omega(\delta(n))$, it holds that

$$T^h_2(n) + T^v_2(n) = O((\log n)^{\frac{\alpha}{2}} \cdot N \cdot (\zeta^h_{max} + \zeta^v_{max})).$$

Combining with Lemma 6, we can prove this theorem. □

From the analysis above, $T^h_2(n)$ and $T^v_2(n)$ depend on the types of aggregation functions indeed. Consequently, we instantiate the general result in Theorem 1 to a special case, i.e., the case of *divisible perfectly compressible functions*.

3.2 Aggregation Throughput for Divisible-Perfectly-Compressible Functions

From the characteristic of *divisible-perfectly-compressible aggregation functions* (DPC-AFs, Lemma 1), by Theorem 1, we have the following result,

Theorem 2. *For DPC-AFs, the achievable aggregation throughput under the scheme $\mathcal{A}_{N,n}$ with $N = \Omega(\frac{\sqrt{n}}{\sqrt{\log n}})$ is $\Lambda(n) = \Omega((\log n)^{-\frac{\alpha}{2}-1})$.*

Proof. By Lemma 1, for DPC-AFs,

$$\zeta^h_{max} \le \max\{\log |\mathcal{G}_{|D^h_{i,j}|}|\} = \Theta(\log \mathfrak{m}).$$

Similarly, $\zeta^v_{max} = O(\log \mathfrak{m})$. Recall that $\mathfrak{m} = \Theta(1)$, the theorem can thus be proven by using Theorem 1. □

3.3 Aggregation Scheme for Type-Threshold DPC-AFs

Sensing measurements are periodically generated, so the function of interest is required to be computed repeatedly. Hence, the technique, called *block coding* [4], is permitted. The technique of block coding combines several consecutive function computations, and can significantly improve the throughput for *type-threshold functions* [4] in the *collocated network* whose interference graph is a complete graph. For a given round of measurements, denoted by a n-vector $M^n \in \mathcal{M}^n$, the max function, the min function and the range function $(\max_i M_i - \min_i M_i)$ of M^n, the kth largest value of M^n, the mean of the k largest values of M^n, and the indicator function $I\{M_i = k, \text{ for some } i\}$ are all type-threshold functions. We first refer to a result of [4] (Part of Theorem 4 in [4]).

Lemma 12. ([4]). *Under the protocol model, the aggregation capacity for type-threshold functions in a collocated network of n vertices is of order $\Theta(1/\log n)$.*

Under our scheme $\mathcal{A}_{N,n}$, in each cell of the scheme lattice $\mathbb{L}_1$, the communication graph can be regarded as a *collocated network* of $\Theta(\log n)$ vertices, because any two links in a cell can not be scheduled simultaneously during the local aggregation phase. Then, it is possible to improve the throughput by introducing the block coding into the scheme $\mathcal{A}_{N,n}$. The main question to be solved is how to extend the result of Lemma 12 to that under the generalized physical model. Analyze the proof of Lemma 12: Let $N = \Theta(n)$, and under the assumption that each successful transmission can achieve a constant rate, prove that it takes $O(n \log n)$ time slots to finish the aggregation for N rounds of measurements. Thus, since during the local aggregation phase of $\mathcal{A}_{N,n}$, each successful transmission can sustain a rate of order $\Omega((\log n)^{-\frac{\alpha}{2}})$ instead of a constant order, we have the following result,

Lemma 13. *Under generalized physical model, by block coding with $N_b = \Theta(\log n)$, for type-threshold DPC-AFs, the time cost of the local aggregation for $N = \Omega(\log n)$ rounds of measurements is of order $T^{bc}_1(n) = O(N \cdot (\log n)^{\frac{\alpha}{2}} \cdot \log \log n)$.*

Proof. For the communication graph in each cell, we implement the local aggregation by using block coding with length $N_b = \Theta(\log n)$. Similar to Lemma 12, the time cost of aggregating N_b rounds of measurements is of $O((\log n)^{\frac{\alpha}{2}} \cdot N_b \cdot \log \log n)$. By partitioning N rounds of measurements into blocks of length N_b, we prove the lemma. □

Lemma 12 holds when $N = \Omega(\log n)$ which does not contradict with the condition that $N = \Omega(\frac{\sqrt{n}}{\sqrt{\log n}})$ in Theorem 1 and Theorem 2. Then, we can modify the scheme $\mathcal{A}_{N,n}$ by introducing the block coding in the local aggregation phase. Denote this scheme by $\mathcal{A}^{bc}_{N,n}$. Finally, we propose,

Theorem 3. *For type-threshold DPC-AFs, the achievable aggregation throughput under the scheme $\mathcal{A}^{bc}_{N,n}$ with $N = \Omega(\frac{\sqrt{n}}{\sqrt{\log n}})$ is of order*

$$\Lambda(n) = \Omega\left((\log n)^{-\frac{\alpha}{2}} \cdot \frac{1}{\log \log n}\right). \tag{13}$$

Proof. By using block coding, $T(\mathcal{A}^{bc}_{N,n}) = T^{bc}_1(n) + T^h_2(n) + T^v_2(n) = O(N \cdot (\log n)^{\frac{\alpha}{2}} \cdot \log \log n)$. According to Definition 1, we can complete the proof. □

4 Upper Bounds on Aggregation Capacity

In this section, we compute the upper bounds on aggregation capacities for *type-sensitive divisible perfectly compressible aggregation functions* (type-sensitive DPC-AFs) and *type-threshold divisible perfectly compressible aggregation functions* (type-threshold DPC-AFs) over RE-WSN.

4.1 Upper Bounds for Type-Sensitive DPC-AFs

Theorem 4. *The aggregation capacity for type-sensitive DPC-AFs over RE-WSN is of order $O((\log n)^{-\frac{\alpha}{2}-1})$.*

Proof. By Equation (2), in any aggregation tree, there exists, *w.h.p.*, a link of length $\Omega(\sqrt{\log n})$, say uv. The capacity of such a link is upper bounded by

$$B\log\left(1+\frac{(\kappa_1\sqrt{\log n})^{-\alpha}}{N_0}\right)=O((\log n)^{-\frac{\alpha}{2}}),$$

where $\kappa_1>0$ is a constant. According to the characteristics of type-sensitive DPC-AFs, it takes at least $\kappa_2\cdot nN\cdot(\log n)^{\frac{\alpha}{2}}$ transmissions to finish the aggregation of N rounds of measurements from every sensor, where $\kappa_2>0$ is a constant that has no impact on the final results in order sense. By a similar procedure to Lemma 3 (based on VC theorem [25]), we get that each cell in the network lattice $\mathbb{L}(\sqrt{n},\frac{2}{\kappa_2}\sqrt{\log n},n)$ must make at least $\kappa_3\log n$ transmissions, where $\frac{1}{2}<\kappa_3<8$. Since the *arena-bounds* [26] for the generalized physical model is of order $O(\log n)$, [27], the total aggregation rate of those $\kappa_3\log n$ transmissions can be upper bounded of order $O(\mu(\log n)^{-\frac{\alpha}{2}})$ when the data from the senders of these $\kappa_3\log n$ transmissions are aggregated into μ receivers, where $\mu=O(\log n)$. For any aggregation tree, consider the cells in $\mathbb{L}(\sqrt{n},\frac{2}{\kappa_2}\sqrt{\log n},n)$, from the farthest (in hop-distance) cell to that contains the sink node, there must be a scenario where $\mu=\Theta(1)$, because all data will converge to the sink node. In this case, $\kappa_3\log n$ transmissions share the total link rate of $O((\log n)^{-\frac{\alpha}{2}})$, and it takes $\Omega(nN(\log n)^{\frac{\alpha}{2}+1})$ slots to finish the aggregation. Thus, the aggregation capacity is bounded by $\frac{nN}{\Omega(nN(\log n)^{\frac{\alpha}{2}+1})}=O((\log n)^{-\frac{\alpha}{2}-1})$. □

4.2 Upper Bounds for Type-Threshold DPC-AFs

Theorem 5. *The aggregation capacity for type-threshold DPC-AFs over RE-WSN is of order* $O(\frac{(\log n)^{-\alpha/2}}{\log\log n})$.

Proof. For type-threshold DPC-AFs, by a similar procedure to Theorem 4 and according to Theorem 4 of [4], each cell in the network lattice $\mathbb{L}(\sqrt{n},\kappa_4\sqrt{\log n},n)$ must make at least $\kappa_5N\log\log n$ transmissions, when each sensor produces N rounds of measurements, where $\kappa_4,\kappa_5>0$ are some constants. By a similar argument to Theorem 4, there must be a level of aggregation that takes at least $\frac{\Omega(N\log\log n)}{O((\log n)^{-\alpha/2})}=\Omega(N\log\log n\cdot(\log n)^{\alpha/2})$, which completes the proof. □

Combining the lower bounds (Theorem 2 and Theorem 3) with upper bounds (Theorem 4 and Theorem 5), we get that

Theorem 6. *The aggregation capacities for type-sensitive DPC-AFs and type-threshold DPC-AFs over RE-WSN are of order* $\Theta((\log n)^{-\frac{\alpha}{2}-1})$ *and* $\Theta(\frac{(\log n)^{-\alpha/2}}{\log\log n})$, *respectively.*

5 Improving Capacity by Multiple Sinks

According to Theorem 1, Theorem 2 and Theorem 3, we learn that even by using block coding, the bottleneck still lies in the local aggregation phase. More specifically, the size of each collocated network, i.e., the number of nodes in each cell, is still too large. Consequently, we resort to reducing the size of the collocated network without diminishing the link rate. As a solution, we exploit a technique called *parallel transmission scheduling* [9].

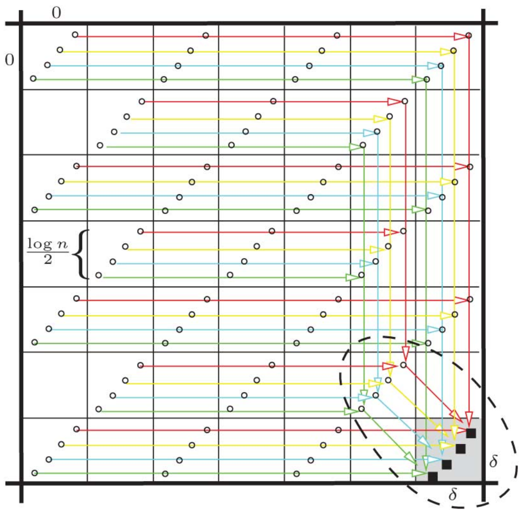


Fig. 5. Parallel aggregation backbones. Black cycles denote aggregation stations. Black squares denote $\Theta(\log n)$ sink nodes that are deployed randomly in a small region (shaded) of area $\Theta(\log n)$. In each row (or column $\delta-1$ and column δ), there are $\frac{\log n}{2}$ horizontal (or vertical) backbones.

5.1 Parallel Aggregation Scheme $\mathcal{A}^p_{N,n}$

According to Lemma 3, in any cell of the network lattice $\mathbb{L}_1$, the number of nodes is *w.h.p.* within $(\frac{1}{2}\log n, 8\log n)$. Hence, intuitively, when we randomly choose $\frac{1}{2}\log n$ stations from each cell in $\mathbb{L}_1$, there are $\frac{1}{2}\log n$ backbones in each row (or column). However, the rate of each backbone does not exceed that of the backbone in the scheme $\mathcal{A}_{N,n}$ because of the nondecreasing interference to each link. Thereby, we attempt to simultaneously schedule multiple backbones in each row (or column), without impairing the rate of every backbone, that is, with sustaining the rate of order $\Omega((\log n)^{-\frac{\alpha}{2}})$. We denote the new scheme with parallel scheduling by $\mathcal{A}^p_{N,n}$. Like $\mathcal{A}_{N,n}$, the scheme $\mathcal{A}^p_{N,n}$ also consists of two phases: *local aggregation phase* and *backbone aggregation phase*; and the backbone phase is further divided into two subphases: *horizontal backbone phase* and *vertical backbone phase*.

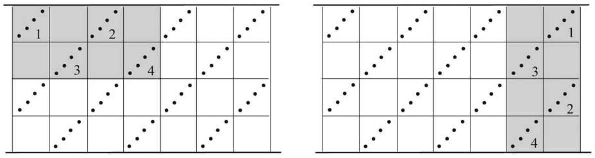
Backbone Aggregation Phase: Denote the stations in the cell $C_{i,j}$ of $\mathbb{L}_1$ by $b_{i,j}(\iota)$, $1\le\iota\le\frac{\log n}{2}$. Then, we construct the horizontal backbones in the following method (Please see illustration in Fig. 5): In each row i, $0\le i\le\delta$,

- when i is even, for $0\le k\le\lfloor\frac{\delta}{2}\rfloor-1$ and $1\le\iota\le\frac{\log n}{2}$, connect the station $b_{i,2k}(\iota)$ to $b_{i,2k+2}(\iota)$;
- when i is odd, for $0\le k\le\lfloor\frac{\delta}{2}\rfloor-1$ and $1\le\iota\le\frac{\log n}{2}$, connect the station $b_{i,2k+1}(\iota)$ to $b_{i,2k+3}(\iota)$.

Similarly, we can build the vertical backbones in column $\delta-1$ and column δ. Note that not all cells contain the station.

For the convenience of description, we assume that δ is even without loss of generality. During the horizontal backbone phase, the data are aggregated into the stations on the vertical backbones in two columns; during the vertical backbone phase, the data are aggregated to the stations in the cells $C_{\delta-1,\delta-1}$ and $C_{\delta,\delta}$. When δ is odd, the data are finally aggregated into the stations in the cells $C_{\delta-1,\delta}$ and $C_{\delta,\delta-1}$.

Now, the key question is how to schedule the backbones to improve the total rate. To schedule the horizontal backbones, we adopt a 4-TDMA scheme, denoted by $H^p_{N,n}$. Under the scheme $H^p_{N,n}$, as illustrated in Fig. 6(a), a *scheduling unit* (SU) consists of 2×4 cells. Since there are only four cells that



(a) horizontal scheduling (b) vertical scheduling

Fig. 6. Parallel backbone scheduling. The $\frac{\log n}{2}$ stations in each scheduled cell can transmit simultaneously.

contain stations and need to be scheduled in each SU, we can complete the scheduling of each SU once in a period of four time slots. Similarly, we can use a 4-TDMA scheme, denoted by $V^p_{N,n}$ and described in Fig. 6(b), to schedule the vertical backbones. we iterate that for both $H^p_{N,n}$ and $V^p_{N,n}$, the key technique to improve the throughput is to permit $\frac{\log n}{2}$ stations in a scheduled cell to transmit simultaneously. Next, we prove the improvement in the total rate of backbones.

Lemma 14. *Under the scheduling scheme $\mathcal{H}^p_{N,n}$ (or $\mathcal{V}^p_{N,n}$), every horizontal (or vertical) backbone can still sustain a rate of order $\Omega((\log n)^{-\frac{\alpha}{2}})$.*

Proof. Let us consider any scheduled link of horizontal backbones. The interferences at the receiver of the link are produced by those $\frac{\log n}{2} - 1$ stations in the cell containing the intended transmitter and ones in other scheduled cells. Since the length of the link is at least of $\mathfrak{l} = 2\sqrt{\log n}$, the sum of interferences at the receiver can be bounded by $I^p_h(n) \leq P \cdot (\frac{\log n}{2} - 1) \cdot \ell(2\sqrt{\log n}) + \sum_{i=1}^n 8iP \cdot \frac{\log n}{2} \cdot \ell((4i-3) \cdot 2\sqrt{\log n}) \leq \frac{P}{2^{\alpha+1}} \cdot (\log n)^{1-\frac{\alpha}{2}} \cdot (1 + \lim_{n\to\infty} \sum_{i=1}^n \frac{8i}{(4i-3)^\alpha})$. The latest limitation converges when $\alpha > 2$. Since the distance of every hop is at most $\sqrt{10} \cdot \mathfrak{l}$, we have the signal at the receiver can be bounded as $S^p_h(n) \geq 40^{-\alpha} P \cdot (\log n)^{-\frac{\alpha}{2}}$. Then, $\frac{S^p_h(n)}{N_0 + I^p_h(n)} = O((\log n)^{-\frac{\alpha}{2}})$. since $\log(1+x) \sim x$ as $x = o(1)$, the scheduled link can achieve the rate of $R^p_h(n) = \Theta((\log n)^{-\frac{\alpha}{2}})$. Since all horizontal backbones are scheduled once in four time slots, every horizontal backbone can sustain a rate of $\frac{R^p_h(n)}{4}$, which proves the result on horizontal backbones. In a similar way, we get the result on vertical backbones. □

Local Aggregation Phase: Now, we discuss how to efficiently collect the measurements into those stations. We still set the scheduling unit as a cluster of 2×4 cells, like in Fig. 6 (a). Denote a scheduling unit by $U_{h,v}$ that consists of the cells $C_{2h,4v}$, $C_{2h,4v+1}$, $C_{2h,4v+2}$, $C_{2h,4v+3}$, $C_{2h+1,4v}$, $C_{2h+1,4v+1}$, $C_{2h+1,4v+2}$, and $C_{2h+1,4v+3}$, if any. The parallel routing and scheduling scheme can be described as follows. To simplify the description, let $t_0 - t_1$: $C_{i,j} \Rrightarrow C_{k,l}$ represent that all measurements from nodes in $C_{i,j}$ are evenly aggregated into the $\frac{\log n}{2}$ stations in $C_{k,l}$ during time slots $t_0 - t_1$. It takes 128 time slots to schedule all links in $U_{h,v}$:

$$\begin{array}{lllllll}
1 & - & 16 & : & C_{2\,h,4v} & \Rrightarrow & C_{2\,h,4v+2} \\
17 & - & 32 & : & C_{2\,h,4v+1} & \Rrightarrow & C_{2\,h+1,4v+3} \\
33 & - & 48 & : & C_{2\,h,4v+2} & \Rrightarrow & C_{2\,h,4v} \\
49 & - & 64 & : & C_{2\,h,4v+3} & \Rrightarrow & C_{2\,h+1,4v+1} \\
65 & - & 80 & : & C_{2\,h+1,4v} & \Rrightarrow & C_{2\,h,4v+2} \\
81 & - & 96 & : & C_{2\,h+1,4v+1} & \Rrightarrow & C_{2\,h+1,4v+3} \\
97 & - & 112 & : & C_{2\,h+1,4v+2} & \Rrightarrow & C_{2\,h,4v} \\
113 & - & 128 & : & C_{2\,h+1,4v+3} & \Rrightarrow & C_{2\,h+1,4v+1}
\end{array}$$

Under the above scheduling scheme, the hop-length of every link is of order $\Omega(\sqrt{\log n})$. From the analysis in Lemma 14, we can schedule simultaneously $\frac{\log n}{2}$ links in each scheduled cell, ensuring that the rate of every link is of order $R^p_1(n) = \Omega((\log n)^{-\alpha/2})$. On the other hand, it takes at most 16 time slots to schedule all links from one cell to the stations in other cell, because $\frac{\log n}{2}$ links can be simultaneously scheduled while there are at most $8 \log n$ nodes in each cell.

5.2 Throughput under Aggregation Scheme $\mathcal{A}^p_{N,n}$

First, for the local aggregation phase, we have the following result,

Lemma 15. *Under the parallel scheduling scheme, the time cost of the local aggregation for N rounds of measurements is of order $T^p_1(n) = O(N \cdot (\log n)^{\frac{\alpha}{2}})$.*

Proof. Since in the local phase, each link can be scheduled once out of 128 time slots, then the average rate of each link can sustain of order $\Lambda^p_1(n) = \frac{R^p_1(n)}{128}$, i.e., $\Lambda^p_1(n) = \Omega((\log n)^{-\frac{\alpha}{2}})$. And, each sensor has a load of N measurements to transmit into the corresponding station, thus, the total time cost is $\frac{N \log \mathfrak{m}}{\Lambda^p_1(n)} = O(N \cdot (\log n)^{\frac{\alpha}{2}})$, which completes the proof. □

Next, we consider the backbone phase. Under the scheme $\mathcal{A}^p_{N,n}$, define the set of $b_{i,j}(\iota)$ and all its descendants as $D^{p,h}_{i,j}(\iota)$ for all i, j and ι; and define the set of $b_{i,j}(\iota)$ and all its descendants as $D^{p,v}_{i,j}(\iota)$ for all i, ι and $j = \delta - 1, \delta$. Since there are $\frac{\log n}{2}$ stations that contribute the burden-sharing of each cell, then it holds that

$$|D^{p,h}_{i,j}(\iota)| = \Theta(\frac{2}{\log n}) \cdot |D^h_{i,j}|; |D^{p,v}_{i,j}(\iota)| = \Theta(\frac{2}{\log n}) \cdot |D^v_{i,\delta}|$$

Then, in the horizontal phase and vertical phase, the load produced by kth round of measurements on station $b_{i,j}(\iota)$ is

$\zeta^{p,h}_{i,j}(\iota) = \log |\mathcal{G}_{|D^{p,h}_{i,j}|}|$ and $\zeta^{p,v}_{i,j}(\iota) = \log |\mathcal{G}_{|D^{p,v}_{i,j}|}|$, respectively. Then, we further define

$$\zeta^{p,h}_{max} := \max_{i,j,\iota} \zeta^{p,h}_{i,j}(\iota), \zeta^{p,v}_{max} := \max_{i,j,\iota} \zeta^{p,v}_{i,j}(\iota). \tag{14}$$

Thus, in a similar procedure to the proofs of Lemma 9 and Lemma 11, we can obtain

Lemma 16. *Under the parallel scheduling scheme, to finish the backbone aggregation for N rounds of measurements, it takes at most*

$$T^{p,h}_2(n) = O((\zeta^{p,h}_{max} + \zeta^{p,v}_{max}) \cdot (\log n)^{\frac{\alpha}{2}} \cdot (N + \delta(n)))$$

seconds, where $\zeta^{p,h}_{max}$ and $\zeta^{p,v}_{max}$ are defined in (14).

Combining Lemma 15 and Lemma 16, we get that,

Theorem 7. *Under the parallel aggregation scheme $\mathcal{A}^p_{N,n}$ with $N = \Omega(\sqrt{n}/\sqrt{\log n})$, the measurements from all n sensors can be aggregated into $\Theta(\log n)$ sink nodes in a small region of which the area is an infinitesimal proportion, i.e., $\frac{4 \log n}{n}$, of the whole deployment region, at a throughput of*

$$\Lambda(n) = \Omega\left(\frac{(\log n)^{-\frac{\alpha}{2}}}{1 + \zeta^{p,h}_{max} + \zeta^{p,v}_{max}}\right). \tag{15}$$

For the DPC-AFs, such achievable throughput is of order $\Lambda(n) = \Omega((\log n)^{-\frac{\alpha}{2}})$.

5.3 Implications of Theorem 7

5.3.1 Elimination of Interference-Limitedness

According to Lemma 2, for any aggregation tree, there must be a link of length $\Omega(\sqrt{\log n})$, which leads to the network throughput is limited to the rate of such a link, i.e., the order $O((\log n)^{-\frac{\alpha}{2}})$ due to *coverage-limitedness* (Section 2.3.1). From Theorem 7, this upper bound can be indeed achieved by choosing randomly $\Theta(\log n)$ sensors from a small region (relative to the whole deployment region) of area $\Theta(\log n)$.

5.3.2 Gains of Multiple Sinks

It is straightforward that choosing multiple sinks can possibly improve the network throughput. While, some potential questions might be asked:

1. *Why not we choose uniformly at random sensors from the whole network as sink nodes?* The reason is two-fold: (1) Since each sink usually connects with a processing station (e.g., computer) via a reliable link (even a wired link), and it should be always guaranteed with a stable power supply, it is easy to understand that the smaller the region contained sink nodes is, the more convenient the network management is. (2) It cannot improve the order of throughput indeed when we choose uniformly $\Theta(\log n)$ sink nodes. An intuitive (but not rigorous) explanation is as following: divide the network into $\Theta(\log n)$ subregions, choose one sensor from each subregion as a sink node that takes charge of data gathering in this subregion. By doing so, the throughput of each subregion is also of the same order as that of the whole region with a single sink node, because the area of the subregions (then, the number of sensors), i.e., $\Theta(\frac{n}{\log n})$, is still too large to ease the bottlenecks in the whole network.
2. *Can we improve further the aggregation throughput by more sink nodes?* It can definitely improve the network throughput when we choose uniformly a large enough number of sensors as sink nodes. An extreme case is that all sensors are chosen as sink nodes. Of course, it makes no sense. It can be straightforwardly inferred that the throughput under such parallel aggregation schemes can be increased by $k(n)$ times when $\Theta(k(n) \cdot \log n)$ sinks are chosen from a subregion of area $\Theta(k(n) \cdot \log n)$, where $k(n) = \Omega(1)$ and $k(n) = O(\frac{n}{\log n})$. Finally, we conjecture that the throughput derived in Theorem 7 is indeed optimal if the number of sink nodes is limited to the number of $O(\log n)$. We leave the rigorous validation to future work.

5.4 Challenges in Practical Implementation

This work falls within the scope of scaling laws, i.e., scaling of the network performance in the limit when the network gets large, [4]. We aim to investigate the qualitative and fundamental properties of WSNs in terms of aggregation capacity, [4], [10], [28], [29]. The proposed schemes provide some architectural guidelines on how to design the practical aggregation schemes for WSNs that scale well. It should be emphasized that when implementing the specific schemes under the proposed architecture, we need to take into account some technique details, e.g., the tuning of many parameters and optimization of some pre-constants in the system throughput, [10], [29].

6 Literature Review

The issue of capacity scaling laws for wireless ad hoc networks [1] has been intensively studied under different models. The first work about aggregation capacity scaling laws of WSN was done by Marco et al. [5]. They considered the capacity of random dense WSNs under the protocol model [1]. In [4], Giridhar and Kumar also focused on dense WSNs, and investigated the more general problem of computing and communicating symmetric functions of the sensor measurements. They showed that for *type-sensitive functions* and *type-threshold functions*, the aggregation capacities for random dense WSNs under the protocol model are of order $\Theta(\frac{1}{\log n})$ and $\Theta(\frac{1}{\log\log n})$, respectively. Ying et al. [30] studied the optimization problem of the total transmission energy for computing a symmetric function, subject to the constraint that the computation is *w.h. p.*, correct. Moscibroda [8] derived the aggregation capacity scaling laws of *divisible perfectly compressible functions* for *worst-case* networks. They showed that under the protocol model and physical model [1], the capacity for *worst-case* networks is of order $\Omega(\frac{1}{n})$ and $\Omega(\frac{1}{(\log n)^2})$, respectively. Zheng et al. [31] proposed an Attribute-aware Data Aggregation mechanism using Dynamic Routing (ADADR) that can make packets with the same attribute convergent as much as possible and therefore improve the efficiency of data aggregation. Under the cooperative paradigm, Zheng and Barton [7] demonstrated that the upper bound of the capacity of data collecting for extended WSNs is of order $\Theta(\log n/n)$ and $\Theta(1/n)$ when operating in fading environments with power path-loss exponents that satisfy $2 < \alpha < 4$ and $\alpha > 4$, respectively. The work considered the aggregation functions without no *in-network aggregation* [32], e.g., *data downloading problem* [4]. It might be interesting to merge this technique into our scheme to improve the results under the cooperative paradigm.

Note that unlike all other works mentioned, including our work, the result in [7] is *information-theoretic* bounds instead of *networking-theoretic* bounds. (Section 2.1). In [33], Wang et al. studied the capacity, delay and their tradeoffs in k-source converge-cast networks. They showed that the essential key controlling the capacity is the redundancy, with inverse relationship. In [29], [34], by introducing the visual MIMO techniques, Fu and Wang et al. designed two different cooperative schemes for static and mobile ad hoc wireless networks, respectively. Servetto [35] constructed a structured class of asymptotically optimal quantizers for the problem of rate/distortion with side information available only at the decoder. He showed that there exist quantizers whose performance comes arbitrarily close to Wyner¡¯s bound [36]. Guleryuz and Kozat [37] studied an important class of sensor networks where the ultimate goal is not necessary to collect each individual measurement but rather a potentially smaller set of statistics. They determined how the flow of information emanating from the sensors should be managed, yielding optimal routing algorithms and jointly optimized networks.

Some work focused on the data transmission in mobile WSNs where the sensors and/or sinks are mobile. In [38], Chen et al. proposed to use multihop forwarding to form a cluster around the expected position of a mobile sink, in order to guarantee packet delay and minimize energy consumption. Park et al. [39] designed a simple route maintaining algorithm to support the sink mobility of conventional routing protocols. The algorithm does not use the flooding method and does not require the information on the geometric location of sensor nodes. In [40], Gao et al. formulated the problem of jointly improving the amount of data collected and reducing the energy consumption as an integer linear programming problem, and proposed a corresponding data collection scheme based on optimizing the assignment of sensor nodes.

7 Conclusion

We emphasize that for random extended WSNs (RE-WSNs), the basic assumption of the protocol model and physical model [1] that any successful transmission can sustain a constant rate is over-optimistic and unpractical. We derive the first result on scaling laws of the aggregation capacity for RE-WSNs under generalized physical model. We show that, for general divisible perfectly compressible aggregation functions (DPC-AFs), the achievable aggregation throughput of RE-WSNs is of order $\Omega((\log n)^{-\frac{\alpha}{2}-1})$; and for type-sensitive DPC-AFs and type-threshold DPC-AFs, aggregation capacities are of order $\Theta((\log n)^{-\frac{\alpha}{2}-1})$ and $\Theta(\frac{1}{\log\log n}\cdot(\log n)^{-\frac{\alpha}{2}})$, respectively. Furthermore, by introducing the parallel transmission scheduling, we prove that the achievable aggregation throughput for DPC-AFs is of order $\Omega((\log n)^{-\frac{\alpha}{2}})$ by choosing $\Theta(\log n)$ sink nodes in a small region.

There are some interesting directions left to study: (1) An interesting future work is to derive the aggregation capacity scaling laws for mobile WSNs consisting of mobile sensors or mobile sinks with energy limitations. (2) It is worth deriving the aggregations capacity of WSNs for more general aggregation functions. (3) It would be interesting to investigate how real-world wireless propagation, such as shadowing, path loss and multipath fading, affects the aggregation capacity of WSNs by introducing more realistic communication models.

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive comments. The research of authors is partially supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61202383, in part by the National Basic Research Program of China (973 Program) under Grant 2010CB328101, in part by the Shanghai Rising-Star Program under Grant 14QA1403700, in part by the Program for New Century Excellent Talents in University (NCET) under Grant NCET-12-0414, in part by the Natural Science Foundation of Shanghai under Grant 12ZR1451200, in part by the Integrated Project for Major Research Plan of the National Natural Science Foundation of China under Grant 91218301, and in part by the Research Fund for the Doctoral Program of Higher Education of China (RFDP) under Grant 20120072120075.

References

[1] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[2] A. Özgür, O. Lévêque, and D. Tse, "Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3549–3572, 2007.

[3] S. Li, Y. Liu, and X.-Y. Li, "Capacity of large scale wireless networks under Gaussian channel model," in *Proc. ACM Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2008, pp. 140–151.

[4] A. Giridhar and P. Kumar, "Computing and communicating functions over sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 755–764, 2005.

[5] D. Marco, E. Duarte-Melo, M. Liu, and D. Neuhoff, "On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data," in *Proc. ACM Int. Conf. Inf. Process. Sens. Netw. (IPSN'03)*, 2003, pp. 1–16.

[6] R. Barton and R. Zheng, "Cooperative time-reversal communication is order-optimal for data aggregation in wireless sensor networks," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT'06)*, 2006, pp. 222–226.

[7] R. Zheng and R. Barton, "Toward optimal data aggregation in random wireless sensor networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM'07)*, 2007, pp. 249–257.

[8] T. Moscibroda, "The worst-case capacity of wireless sensor networks," in *Proc. ACM/IEEE Int. Conf. Inf. Process. Sens. Netw. (IPSN'07)*, 2007, pp. 1–10.

[9] C. Wang, X.-Y. Li, and C. Jiang, "Scaling laws on multicast capacity of large scale wireless networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM'09)*, 2009, pp. 1863–1871.

[10] C. Wang, C. Jiang, S. Tang, and X.-Y. Li, "Selectcast: Scalable data aggregation scheme in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 10, pp. 1958–1969, 2012.

[11] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin, "Data-centric storage in sensornets," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, p. 142, 2003.

[12] C. Chau, M. Chen, and S. Liew, "Capacity of large-scale CSMA wireless networks," in *Proc. ACM Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2009, pp. 97–108.

[13] M. Franceschetti, O. Dousse, D. Tse, and P. Thiran, "Closing the gap in the capacity of wireless networks via percolation theory," *IEEE Trans. Inf. Theory*, vol. 53, no. 3, pp. 1009–1018, 2007.

[14] R. Zheng, "Asymptotic bounds of information dissemination in power-constrained wireless networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 1, pp. 251–259, Jan. 2008.

[15] O. Dousse and P. Thiran, "Connectivity vs capacity in dense ad hoc networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2004, pp. 476–486.

[16] M. Penrose, "The longest edge of the random minimal spanning tree," *Ann. Appl. Probab.*, vol. 7, pp. 340–361, 1997.

[17] P. Gupta and P. Kumar, "Critical power for asymptotic connectivity in wireless networks," *Stochastic Anal. Control Optim. Appl.: A Volume Honor Fleming*, vol. 3, no. 20, pp. 547–566, 1998.

[18] P. Santi and D. Blough, "The critical transmitting range for connectivity in sparse wireless ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 2, no. 1, pp. 25–39, 2003.

[19] P. Santi, "The critical transmitting range for connectivity in mobile ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 4, no. 3, pp. 310–317, 2005.

[20] X.-Y. Li, "Multicast capacity of wireless ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 950–961, 2009.

[21] C. Wang, C. Jiang, X. Li, and Y. Liu, "Asymptotic throughput for large-scale wireless networks with general node density," *Wireless Netw.*, vol. 19, no. 5, pp. 713–725, 2013.

[22] E. Duarte-Melo and M. Liu, "Data-gathering wireless sensor networks: organization and capacity," *Comput. Netw.*, vol. 43, no. 4, pp. 519–537, 2003.

[23] H. Gamal, "On the scaling laws of dense wireless sensor networks: the data gathering channel," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 1229–1234, 2005.

[24] B. Liu, P. Thiran, and D. Towsley, "Capacity of a wireless ad hoc network with infrastructure," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (Mobihoc)*, 2007, pp. 239–246.

[25] V. Vapnik and A. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities,"*Theory Probab. Its Appl.*, vol. 16, no. 2, pp. 264–280, 1971.

[26] A. Keshavarz-Haddad and R. Riedi, "Bounds for the capacity of wireless multihop networks imposed by topology and demand," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2007, pp. 256–265.

[27] A. Keshavarz-Haddad and R. Riedi, "Multicast capacity of large homogeneous multihop wireless networks," in *Proc. IEEE Int. Symp. Modeling Optim. Mobile Ad Hoc Wireless Netw. (WiOpt)*, 2008, pp. 116–124.

[28] C. Wang, C. Jiang, Y. Liu, X.-Y. Li, S. Tang, and H. Ma, "Aggregation capacity of wireless sensor networks: Extended network case," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2011, pp. 1701–1709.

[29] L. Fu, Y. Qin, X. Wang, and X. Liu, "Throughput and delay analysis for convergecast with MIMO in wireless networks,"*IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 4, pp. 768–775, 2012.

[30] L. Ying, R. Srikant, and G. Dullerud, "Distributed symmetric function computation in noisy wireless sensor networks,"*IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4826–4833, 2007.

[31] J. Zhang, F. Ren, T. He, and C. Lin, "Attribute-aware data aggregation using dynamic routing in wireless sensor networks," in *Proc. IEEE Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM)*, 2010, pp. 1–9.

[32] P. Wan, C. Scott, L. Wang, Z. Wan, and X. Jia, "Minimum-latency aggregation scheduling in multihop wireless networks," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2009, pp. 185–194.

[33] X. Wang, L. Fu, X. Tian, Y. Bei, Q. Peng, X. Gan, H. Yu, and J. Liu, "Converge cast: On the capacity and delay tradeoffs, "*IEEE Trans. Mobile Comput.*, vol. 11, no. 6, pp. 970–982, June 2012.

[34] L. Fu, Y. Qin, X. Wang, and X. Liu, "Converge-cast with MIMO," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2011, pp. 649–657.

[35] S. Servetto, "Lattice quantization with side information: Codes, asymptotics, and applications in sensor networks," *IEEE Trans. Inf. Theory*, vol. 53, no. 2, pp. 714–731, 2007.

[36] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. 22, no. 1, pp. 1–10, 1976.

[37] O. Guleryuz and U. Kozat, "Joint compression, detection, and routing in capacity contrained wireless sensor networks," in *Proc. IEEE/SP 13th Workshop Stat. Signal Process.*, 2005, pp. 1026–1031.

[38] C. Chen, J. Ma, and K. Yu, "Designing energy-efficient wireless sensor networks with mobile sinks," in *Proc. ACM Sensys'06 Workshop WSW*, 2006.

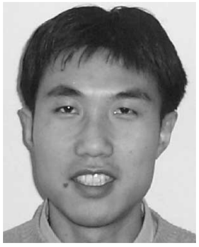
[39] C. Park, K. Lee, Y. Kim, and S. Ko, "A route maintaining algorithm using neighbor table for mobile sinks," *Wireless Netw.*, vol. 15, no. 4, pp. 541–551, 2009.

[40] S. Gao, H. Zhang, and S. Das, "Efficient data collection in wireless sensor networks with path-constrained mobile sinks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 4, pp. 592–608, Apr. 2011.

Cheng Wang received the PhD degree in Department of Computer Science, Tongji University, Shanghai, China, in 2011. Currently, he is with the Department of Computer Science and Engineering, Tongji University. His research interests include wireless communications and networking, mobile social networks, and mobile cloud computing.

Changjun Jiang received the PhD degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1995. Currently, he is a professor with the Department of Computer Science and Engineering, Tongji University, Shanghai, China. His current areas of research are concurrent theory, Petri net, and formal verification of software, wireless networks, concurrency processing, and intelligent transportation systems.

Yunhao Liu (SM'06) received the BS degree in Automation Department from Tsinghua University, China, in 1995, and the MA degree in Beijing Foreign Studies University, China, in 1997, and an MS and a Ph.D. degree in computer science and engineering from Michigan State University, East Lansing, MI, USA, in 2003 and 2004, respectively. He is a member of Tsinghua National Lab for Information Science and Technology, China, and the director of Tsinghua National MOE Key Lab for Information Security, China. He is also a faculty at the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, China. He is also the ACM Distinguished Speaker.



Xiang-Yang Li (M'99–SM'08) received MS and PhD degrees at Department of Computer Science from University of Illinois, Urbana-Champaign, in 2000. He received the Bachelor degree at Department of Computer Science and Bachelor degree at Department of Business Management from Tsinghua University, China, both in 1995. He is an associate professor of computer science at the Illinois Institute of Technology, Chicago, IL, USA. He serves as an editor of of several journals, including *IEEE TPDS* and *Networks*. He also serves in Advisory Board of *Ad Hoc & Sensor Wireless Networks* from 2005, and *IEEE CN* from 2011.

Shaojie Tang has been a PhD student of Computer Science Department, the Illinois Institute of Technology, Chicago, IL, USA, since 2006. He received BS degree in radio engineering from Southeast University, China, in 2006. His current research interests include algorithm design and analysis for wireless ad hoc networks, wireless sensor networks, and online social networks.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.