Snapshot/Continuous Data Collection Capacity for Large-Scale Probabilistic Wireless Sensor Networks

Shouling Ji
Department of Computer Science
Georgia State University
Atlanta, Georgia 30303, USA
Email: sji@cs.gsu.edu

Raheem Beyah
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30308, USA
Email: rbeyah@ece.gatech.edu

Zhipeng Cai
Department of Computer Science
Georgia State University
Atlanta, Georgia 30303, USA
Email: zcai@cs.gsu.edu

***Abstract*—*Data collection* is a common operation of Wireless Sensor Networks (WSNs). The performance of data collection can be measured by its achievable *network capacity*. Most of the current works on the network capacity issue are based on the *deterministic network model*, which is not practical for real applications due to the "*transitional region phenomenon*" [22]. The *probabilistic network model* is actually a more practical one. In this paper, we investigate the achievable Snapshot/Continuous Data Collection (SDC/CDC) capacity for WSNs under the probabilistic network model. For SDC, we propose a novel *Cell-based Multi-Path Scheduling* (CMPS) algorithm, whose achievable network capacity is $\Omega(\frac{p_o}{3\omega}\cdot W)$ in the worst case and $\Omega(\frac{p_o}{\omega}\cdot W)$ in the average case, where p_o is the *promising transmission threshold probability*, ω is a constant, and W is the data transmitting rate over a wireless channel, *i.e.* the channel bandwidth, which are both order-optimal. For CDC, we propose a *Zone-based Pipeline Scheduling* (ZPS) algorithm. ZPS significantly speeds up the data collection process and achieves surprising network capacities for both the worst case and the average case. The simulation results also validate that the proposed algorithms significantly improve network capacity compared with the existing works.**

I. Introduction

Wireless Sensor Networks (WSNs) are mainly used for collecting data from the physical world. Generally, data gathering can be categorized as *data aggregation* [10], which obtains aggregated values, *e.g.* the maximum, minimum, and/or average values of all the data, and *data collection* [4][8][15], which gathers all the data from the network without any data aggregation. For data collection, the union of all the values from all the sensors at a particular time instant is called a *snapshot* [4]. The problem of collecting one snapshot is called *Snapshot Data Collection* (SDC). On the other hand, the problem of collecting multiple continuous snapshots is called *Continuous Data Collection* (CDC). To evaluate network performance, *network capacity*, which can reflect the achievable data transmission/collection rate, is usually used [1], [2], [4], [5], [8], [12], [18], [20]. Particularly, for unicast, multicast, and broadcast, we use *unicast capacity*, *multicast capacity*, and *broadcast capacity* to denote the network capacity, respectively. For data collection, we use the data receiving rate at the sink, referred to as *data collection capacity*, to measure its achievable network capacity, *i.e.* data collection capacity reflects how fast data has been collected at the sink[1].

After the seminal work [7], many works emerged to study the network capacity issue under the *Protocol Interference Model* (PrIM) [4][6][8][19][21] or/and the *Physical Interference Model* (PhIM) [3][11][21] for a variety of network scenarios, *e.g.* multicast capacity [12], unicast capacity [16], broadcast capacity [13], and SDC capacity [4][15]. All of the above mentioned works are based on the *Deterministic Network Model* (DNM), where any pair of nodes in a network is either *connected* or *disconnected*. If two nodes are connected, *i.e.* there is a deterministic link between them, then a successful data transmission can be guaranteed as long as there is no collision. For the WSNs considered under the DNM, we call them *deterministic WSNs*. However, in real applications, this DNM assumption is too ideal and not practical due to the "*transitional region phenomenon*" [14][22]. With the transitional region phenomenon, a large number of network links (more than 90% [14]) become unreliable links, named *lossy links* [14]. Even without interference, data transmission over a lossy link is successfully conducted with a certain probability, rather than being completely guaranteed. Therefore, a more practical network model for WSNs is the *Probabilistic Network Model* (PNM) [14], in which data communication over a link is successful with a certain probability rather than always successful or always failing. For convenience, the WSNs considered under the PNM are called *probabilistic WSNs*.

Unfortunately, for the network capacity (including unicast, multicast, broadcast and data collection capacities) issues, to the best of our knowledge, all of the existing works are based on the ideal DNM rather than the more realistic PNM. This motivates us to investigate the achievable network capacity of WSNs under the PNM. Specifically, in this paper, we study the achievable SDC and CDC capacity for probabilistic WSNs. We first investigate how to partition a probabilistic WSN into *cells* and *zones* to improve the concurrency of the data collection process. Subsequently, we propose two data collection schemes, the *Cell-based Multi-Path Scheduling* (CMPS) algorithm and the *Zone-based Pipeline Scheduling*

[1]Without confusion, we use data collection capacity and network capacity interchangeably in the following of this paper.

(ZPS) algorithm for SDC and CDC respectively. Particularly, the main contributions of this paper are summarized as follows:

- For a probabilistic WSN deployed in a square area, we first partition the network into small *cells*. Then, we abstract each cell to a *super node* in the *data collection tree* built for data collection. Based on the data collection tree, we design a novel *Cell-based Multi-Path Scheduling* (CMPS) algorithm for SDC. Theoretical analysis shows that the achievable network capacity of CMPS is $\Omega(\frac{p_o}{3\omega} \cdot W)$ in the worst case and $\Omega(\frac{p_o}{\omega} \cdot W)$ in the average case, where p_o is the *promising transmission threshold probability* defined in Section II, ω is a constant defined in Section III, and W is the data transmitting rate over a wireless channel, *i.e.* the channel bandwidth. Since the upper bound of the SDC capacity is shown to be W [4][8], CMPS successfully achieves the order-optimal network capacity, which is the first technique and currently the best one to address this problem for probabilistic WSNs.
- By combining the CDG technique [15] and the pipeline technique, we propose a novel *Zone-based Pipeline Scheduling* (ZPS) algorithm for continuous data collection in probabilistic WSNs. Taking the benefits brought by CDG and pipeline, ZPS improves the achievable network capacity significantly. For collecting N continuous snapshots, we theoretically prove that the asymptotic achievable network capacity of ZPS is $\sqrt{n/\log n}$ or $n/\log n$ times better than the optimal capacity in order, which is a significant improvement.
- The simulation results also indicate that the proposed algorithms significantly improve the network capacity compared with existing works for probabilistic WSNs.

The rest of this paper is organized as follows: Section II and Section III introduce the probabilistic network model and the network partition strategy which is crucial for the proposed data collection methods, respectively. The Cell-based Multi-Path Scheduling (CMPS) algorithm for snapshot data collection in probabilistic WSNs is proposed and analyzed in Section IV. Section V presents a novel Zone-based Pipeline Scheduling (ZPS) scheme for continuous data collection and its theoretical achievable asymptotic network capacity is shown. Extensive simulations to validate the proposed algorithms are conducted in Section VI and we conclude this paper in Section VII.

II. Network Model

In this paper, we consider a WSN consisting of n sensors, denoted by $s_1, s_2, \cdots, s_n$ respectively, and one sink deployed in a square plane with area $A = cn$ (*i.e.*, the node density of the network is $1/c$), where c is a constant. Furthermore, we assume the distribution of all the sensors is independent and identically distributed (i.i.d.) and loosing only a constant factor, the sink is located at the top-right corner of the square[2].

[2]When the sink is in the middle, one achieves 1/4 of capacity of the sink in the corner.

At each time interval, every sensor generates a data packet with size B bits, and transmits its data to the sink via a multi-hop path over a common wireless channel with bandwidth W bits/second, *i.e.* the data transmission rate of the common channel is W. We further assume the time is slotted into time slots with each of length $t_o = B/W$ seconds.

During the data collection process, all the sensors in the network transmit data with a fixed power P. Therefore, when sensor s_i transmits a packet to sensor s_j, the SINR (Signal-to-Interference-and-Noise-Ratio) associated with s_i at s_j is defined as

$$\Lambda(s_i, s_j) = SINR(s_i, s_j) = \frac{P \cdot \|s_i - s_j\|^{-\alpha}}{N_0 + \sum_{k \neq i} P \cdot \|s_k - s_j\|^{-\alpha}}, \tag{1}$$

where $\|s_i - s_j\|$ is the Euclidean distance between s_i and s_j, α is the path-loss exponent and usually $\alpha \in [3, 5]$, $N_0 > 0$ is a constant representing the background noise, and s_k is another concurrent sender when s_i is transmitting data to s_j. Traditionally, in a DNM, people assume that if the SINR value at a node is greater than or equal to a threshold value, the packet can be received successfully. However, in real applications, due to many *lossy links*, this DNM is too ideal. To be more practical and realistic, instead of employing the DNM, we define a *probabilistic network model*, where each link is associated with a *success probability* which indicates the probability that a successful data transmission can be conducted over this link. Based on the empirical literatures [22], we define the success probability associated with s_i and s_j as

$$\Pr(s_i, s_j) = (1 - \eta_1 \cdot e^{-\eta_2 \cdot \Lambda(s_i, s_j)})^{\eta_3}, \tag{2}$$

where η_1, η_2 and $\eta_3 > 1$ are positive constants. Clearly, when s_i transmits a data packet to s_j, until a successful transmission (*i.e.*, s_j successfully received the whole data packet), the number of transmission times satisfies the *geometric distribution* with parameter $\Pr(s_i, s_j)$. Therefore, the expected transmission times from s_i to s_j is $1/\Pr(s_i, s_j)$, *i.e.* this transmission takes $1/\Pr(s_i, s_j)$ time slots.

The success probability should not be too low, which may imply many transmission times. Therefore, we introduce a *promising transmission threshold probability* p_o. For any promising transmission, we require its success probability to be no less than p_o, *i.e.* for any node pair s_i and s_j, the transmission between s_i and s_j can be conducted only if $\Pr(s_i, s_j) \geq p_o$. Now, for any qualified communication pair to transmit one data packet, the expected transmission time is no more than t_o/p_o. For convenience, we define a *modified time slot* $t_m = t_o/p_o$.

We formally define the achievable *data collection capacity* as the ratio between the amount of data successfully collected by the sink and the time Γ used to collect these data. For instance, to collect N continuous snapshots, the achievable data collection capacity is defined as NnB/Γ, which is actually the data receiving rate at the sink. Particularly, when $N = 1$, nB/Γ is the SDC capacity.

III. Network Partition

In this section, we explain the network partition method, which is essential for our following data collection method.

A. Cell-Based Network Partition

We assume the network is distributed in a square with area size $A = cn$. Now, we partition the network into small square *cells* with edge length $l = \sqrt{2c \log n}$ by a group of horizontal and vertical lines. The resulting network is shown in Fig.1 (a). For convenience, we use $m = \sqrt{cn}/\sqrt{2c \log n} = \sqrt{n/2 \log n}$ to denote the number of cells in each column/row and further define $m' = m - 1$. For each cell shown in Fig.1 (a), we assign each cell a pair of integer coordinates (i, j) $(1 \leq i, j \leq m)$, and a cell with coordinates (i, j) is denoted by $\kappa_{i,j}$. Clearly, the sink is located at the cell $\kappa_{m,m}$. Based on the network partition method, and considering that the sink is located at the top-right corner cell, we decide the possible communication modes for each cell (actually, for the sensors in each cell)[3] as *upward transmission*, *rightward transmission*, and *up-rightward transmission*. Taking cell $\kappa_{i,j}$ as an example, when $\kappa_{i,j}$ works in the upward (rightward/up-rightward) transmission mode, it transmits its data to cell $\kappa_{i,j+1}$ ($\kappa_{i+1,j}$/$\kappa_{i+1,j+1}$).

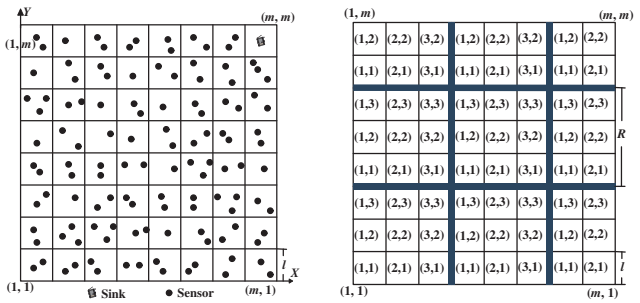

Fig. 1. (a) Network partition and (b) equivalence classes (CTCSs) and zones.

For cell $\kappa_{i,j} (1 \leq i, j \leq m)$, let the random variable $\chi_{i,j}$ denote the number of sensors in it. Then, based on the above network partition, the following three lemmas can be derived and their proofs are omitted due to space limitations.

Lemma 1: The expected number of sensors $E[\chi_{i,j}]$, i.e., the average number of sensors, in $\kappa_{i,j}$ $(1 \leq i, j \leq m)$ is $2 \log n$.

Lemma 2: For cell $\kappa_{i,j}$, let $e_{i,j}$ denote the event that $\kappa_{i,j}$ is empty, i.e., $\chi_{i,j} = 0$. Then $\Pr(e_{i,j}) = \Pr(\chi_{i,j} = 0) \leq \min_{\xi<0} \exp(2(e^{\xi} - 1) \log n - \xi)$. Particularly, $\Pr(\chi_{i,j} = 0) \leq \frac{e^2}{n^{12/7}}$ when $\xi = -2$.

Lemma 3: For any cell $\kappa_{i,j}$, $\Pr(\kappa_{i,j}$ contains more than $6 \log n$ sensors$) = \Pr(\chi_{i,j} > 6 \log n) \leq \frac{1}{n^2}$, i.e., it is almost sure that $\kappa_{i,j}$ contains no more than $6 \log n$ sensors.

From Lemma 1 we know that the average number of sensors in a cell is $2 \log n$. Lemma 2 implies that for large WSNs (large n), every cell contains some sensors. Furthermore, from Lemma 3, it is reasonable to use $6 \log n$ as the upper bound of the number of sensors in a cell. In the following discussion, we assume a cell contains $2 \log n$ sensors in the *average case* and $6 \log n$ sensors in the *worst case*.

[3]To minimize confusion, we use cell and the sensors within this cell interchangeably.

B. Zone-Based Network Partition

After partitioning the network into cells, we want to find which cells can carry out transmissions concurrently. Further, for these cells that can conduct transmissions concurrently, we define them as a *Compatible Transmission Cell Set* (CTCS), denoted by $\mathbb{S}$. Formally, we define $\mathbb{S} = \{\kappa_{i_1,j_1}, \kappa_{i_2,j_2}, \cdots, \kappa_{i_g,j_g} |$ (1) $1 \leq i_k, j_k \leq m$ for $1 \leq k \leq g$; (2) κ_{i_k,j_k} $(1 \leq k \leq g)$ can conduct transmissions concurrently; (3) For κ_{i_k,j_k} $(1 \leq k \leq g)$, suppose κ'_{i_k,j_k} is its destination, *i.e.* κ_{i_k,j_k} transmits data to κ'_{i_k,j_k}, then when κ_{i_k,j_k} $(1 \leq k \leq g)$ conduct transmissions simultaneously, $\min_{1 \leq k \leq g} \Pr(\kappa_{i_k,j_k}, \kappa'_{i_k,j_k}) = \min_{1 \leq k \leq g} \min\{\Pr(s_u, s'_u) | s_u$ is a sensor in κ_{i_k,j_k}, and s'_u is a sensor/sink in $\kappa'_{i_k,j_k}\} \geq p_o.\}$. Clearly, a CTCS is an *equivalence relation* defined on the cells (CTCS is reflexive, symmetric, and transitive). Hence, a CTCS can be viewed as an *equivalence class*.

In order to partition the cells of a WSN into equivalence classes, *i.e.* CTCSs, we assign each cell $\kappa_{i,j}$ $(1 \leq i, j \leq m)$ a vector representation $\vec{\kappa_{i,j}} = ((i-1) \cdot l,\ (j-1) \cdot l) = \kappa_{i,j}$. We further introduce two vectors $\vec{X} = (R, 0)$ and $\vec{Y} = (0, R)$, where $R = \omega \cdot l$, $\omega \in \mathbb{Z}^+$. Then, for any cell $\kappa_{i,j}$ $(1 \leq i,\ j \leq m)$, we define the equivalence class, *i.e.* the CTCS, containing $\kappa_{i,j}$ as the set $\mathbb{S}_{i,j} = \{\vec{\kappa_{i,j}} + a \cdot \vec{X} + b \cdot \vec{Y} | a, b \in \mathbb{Z}\}$, *i.e.* $\mathbb{S}_{i,j} = \{\kappa_{i+a \cdot \omega, j+b \cdot \omega} | a,\ b \in \mathbb{Z}, 1 \leq i + a \cdot \omega, j + b \cdot \omega \leq m\}$ (Here, we suppose $\mathbb{S}_{i,j}$ is a CTCS. Later we will show how to choose R to make it an actual CTCS.). Evidently, $R \geq 3l$ to avoid collision/interference. Taking the WSN shown in Fig.1 (a) as an example, if we set $\omega = 3$, *i.e.* $R = 3l$, then the network can be partitioned into 9 equivalence classes, *i.e.* CTCSs, $\mathbb{S}_{i,j}$ $(1 \leq i, j \leq 3)$ as shown in Fig.1 (b). In Fig.1 (b), the CTCS containing $\kappa_{1,1}$ is $\mathbb{S}_{1,1} = \{\kappa_{1,1}, \kappa_{4,1}, \kappa_{7,1}, \kappa_{1,4}, \kappa_{4,4}, \kappa_{7,4}, \kappa_{1,7}, \kappa_{4,7}, \kappa_{7,7}\}$. Now, we start from the bottom-left corner of the WSN and partition the network into square zones, named *compatible zones*, with edge length $R = \omega \cdot l$ as shown in Fig.1 (b) (where $\omega = 3$). Similarly, for each compatible zone, we use $o_{i,j}$ $(1 \leq i, j \leq \lceil m/\omega \rceil)$ to denote it, and the bottom-left zone with the smallest i and j is $o_{1,1}$. Clearly, $o_{i,j} = \{\kappa_{i',j'} | (i-1) \cdot \omega + 1 \leq i' \leq i \cdot \omega, (j-1) \cdot \omega + 1 \leq j' \leq j \cdot \omega\}$, *i.e.* within a compatible zone, none of the cells belong to the same equivalence class. Furthermore, all the cells with the same relative position in different compatible zones belong to the same equivalence, *i.e.* the same CTCS.

Now, to make any $\vec{X}, \vec{Y}$-based cell set $\mathbb{S}_{i,j}$ an actual CTCS, we need to decide the value of R. Let r be the maximum distance between any communication pair under any communication mode. Then, $r \leq 2\sqrt{2}l$. For large WSNs, the value of R is determined by the following theorem.

Theorem 1: Let $R = \omega \cdot l$, $\omega = \Theta(\frac{r + o(1)}{l})$, $r \leq 2\sqrt{2}l$, $\vec{X} = (R, 0)$, $\vec{Y} = (0, R)$, then the set $\mathbb{S}_{i,j} = \{\vec{\kappa_{i,j}} + a \cdot \vec{X} + b \cdot \vec{Y} | a,$

$b \in \mathbb{Z}\} = \{\kappa_{i+a\cdot\omega,j+b\cdot\omega} | a, b \in \mathbb{Z}, 1 \le i+a\cdot\omega, j+b\cdot\omega \le m\}$ is a CTCS.

Before proving Theorem 1, we prove Lemma 4, which states the lower bound of $\Lambda(\cdot,\cdot)$, and Lemma 5, which shows the lower bound of R, first. In the following proof, assume all the cells in a CTCS $\mathbb{S}_{i,j}$ conduct transmissions concurrently, and all the other cells keep quiet or receive data from some cell in $\mathbb{S}_{i,j}$. When prove Lemma 4, we exploit the similar technique used in [23].

Lemma 4: For each $\mathbb{S}_{i,j}$, $\forall \kappa_{i,j} \in \mathbb{S}_{i,j}$, suppose $\kappa'_{i,j}$ is the destination cell of $\kappa_{i,j}$, then $\Lambda(\kappa_{i,j}, \kappa'_{i,j}) = \min\{\Lambda(s_u, s_v) | 1 \le u, v \le n, s_u \in \kappa_{i,j}, s_v \in \kappa'_{i,j}\} \ge \frac{P\cdot r^{-\alpha}}{N_0 + P\cdot\beta\cdot R^{-\alpha}}$, where $r \le 2\sqrt{l}$ and β is a positive constant.

Proof Sketch: For an arbitrary cell $\kappa_{i,j} \in \mathbb{S}_{i,j}$, suppose it is located in zone $o_{i',j'}$ as shown in Fig.2 (a). Let $\vec{h} = \vec{\kappa_{i,j}}$, then the compatible cells of $\kappa_{i,j}$ in $\mathbb{S}_{i,j}$ can be partitioned into eight disjointed subsets, denoted by A_k $(1 \le k \le 8)$, as shown in Fig.2 (a), where each A_k is defined as follows:

$$\begin{cases} A_1 \equiv \{\vec{h} + b\cdot\vec{Y} | b \in \mathbb{Z}^+\} \equiv \{\kappa_{i,j+b\cdot\omega} | b \in \mathbb{Z}^+\} \\ A_2 \equiv \{\vec{h} + a\cdot\vec{X} | a \in \mathbb{Z}^+\} \equiv \{\kappa_{i+a\cdot\omega,j} | a \in \mathbb{Z}^+\} \\ A_3 \equiv \{\vec{h} + b\cdot\vec{Y} | b \in \mathbb{Z}^-\} \equiv \{\kappa_{i,j+b\cdot\omega} | b \in \mathbb{Z}^-\} \\ A_4 \equiv \{\vec{h} + a\cdot\vec{X} | a \in \mathbb{Z}^-\} \equiv \{\kappa_{i+a\cdot\omega,j} | a \in \mathbb{Z}^-\} \\ A_5 \equiv \{\vec{h} + a\cdot\vec{X} + b\cdot\vec{Y} | a, b \in \mathbb{Z}^+\} \\ \quad \equiv \{\kappa_{i+a\cdot\omega,j+b\cdot\omega} | a, b \in \mathbb{Z}^+\} \\ A_6 \equiv \{\vec{h} + a\cdot\vec{X} + b\cdot\vec{Y} | a \in \mathbb{Z}^+, b \in \mathbb{Z}^-\} \\ \quad \equiv \{\kappa_{i+a\cdot\omega,j+b\cdot\omega} | a \in \mathbb{Z}^+, b \in \mathbb{Z}^-\} \\ A_7 \equiv \{\vec{h} + a\cdot\vec{X} + b\cdot\vec{Y} | a, b \in \mathbb{Z}^-\} \\ \quad \equiv \{\kappa_{i+a\cdot\omega,j+b\cdot\omega} | a, b \in \mathbb{Z}^-\} \\ A_8 \equiv \{\vec{h} + a\cdot\vec{X} + b\cdot\vec{Y} | a \in \mathbb{Z}^-, b \in \mathbb{Z}^+\} \\ \quad \equiv \{\kappa_{i+a\cdot\omega,j+b\cdot\omega} | a \in \mathbb{Z}^-, b \in \mathbb{Z}^+\} \end{cases} . \quad (3)$$

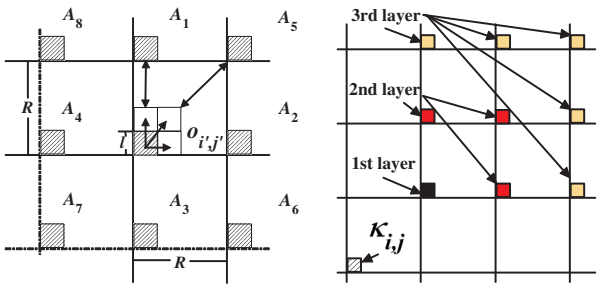

Fig. 2. (a) Interference areas and (b) cell layered of A_5.

Then, for any sensor (sender) s_u in $\kappa_{i,j}$ and its corresponding receiver s_v under any communication mode, we consider the achievable $\Lambda(s_u, s_v)$. Evidently, $\|s_u - s_v\| \le r$ under any communication mode. Furthermore, since P is fixed and N_0 is a constant, the value of $\Lambda(s_u, s_v)$ depends on $\sum_{w\ne u} \|s_w - s_v\|^{-\alpha}$ only. Considering that $\mathbb{S}_{i,j} \setminus \{\kappa_{i,j}\}$ has been partitioned into $A_k (1 \le k \le 8)$, we consider $s_w \in A_k$ $(1 \le k \le 8)$ separately. In the following derivation, we use the facts that $R \ge 3l$, and $\alpha \in [3, 5]$.

Case 1: $s_w \in A_1$. In this case, we have $\|s_w - s_v\| \ge b\cdot R - 2l$, which implies

$$\sum_{s_w\in A_1} \|s_w - s_v\|^{-\alpha} \quad (4)$$
$$\le \sum_{b\ge 1} (b\cdot R - 2l)^{-\alpha} \quad (5)$$
$$\le R^{-\alpha}\cdot[(1-\frac{2}{3})^{-\alpha} + \sum_{b>1}(b-1)^{-2}] \quad (6)$$
$$= R^{-\alpha}\cdot((\frac{1}{3})^{-\alpha} + \frac{\pi^2}{6}) \quad (7)$$
$$= c_1\cdot R^{-\alpha}, \quad (8)$$

where $c_1 = (\frac{1}{3})^{-\alpha} + \frac{\pi^2}{6}$. Similarly, we can prove that for Case g $(2 \le g \le 4, s_w \in A_g)$, $\sum_{s_w\in A_g} \|s_w - s_v\|^{-\alpha} \le c_g \cdot R^{-\alpha}$, where $c_g (2 \le g \le 4)$ are some positive constants.

Case 5: $s_w \in A_5$. In this case, the cells in A_5 can be layered with respect to $\kappa_{i,j}$ with the δ-th layer having $2\delta - 1$ cells[4] as shown in Fig.2 (b). Furthermore, the distance between s_v and any sensor s_w in the δ-th layer is greater than $\delta R - 2l$, *i.e.* $\|s_w - s_v\| \ge \delta R - 2l$ for s_w located in the cell at the δ-th layer. Hence

$$\sum_{s_w\in A_5} \|s_w - s_v\|^{-\alpha} \quad (9)$$
$$\le \sum_{\delta\ge 1} (2\delta - 1)(\delta R - 2l)^{-\alpha} \quad (10)$$
$$\le R^{-\alpha}\cdot(3^\alpha + \sum_{\hat{\delta}\ge 1} 2\hat{\delta}^{-2} + \sum_{\hat{\delta}\ge 1} \hat{\delta}^{-3}) \quad (11)$$
$$= R^{-\alpha}\cdot(3^\alpha + 2\zeta(2) + \zeta(3)) \quad (12)$$
$$= c_5 R^{-\alpha}, \quad (13)$$

where $\zeta(\cdot)$ is the *Riemann zeta function* and $\zeta(2) = \frac{\pi^2}{6}$, $\zeta(3) \approx 1.202$. In the derivation, $\hat{\delta} = \delta - 1$.

Similarly, we can prove that for Case g $(6 \le g \le 8, s_w \in A_g)$, $\sum_{s_w\in A_g} \|s_w - s_v\|^{-\alpha} \le c_g \cdot R^{-\alpha}$, where $c_g (6 \le g \le 8)$ is a positive constant. In a sum, $\sum_{w\ne u} \|s_w - s_v\|^{-\alpha} = \sum_{g=1}^{8} \sum_{s_w\in A_g} \|s_w - s_v\|^{-\alpha} \le \sum_{g=1}^{8} c_g \cdot R^{-\alpha}$. Let $\beta = \sum_{g=1}^{8} c_g$, we have $\Lambda(s_u, s_v) \ge \frac{P\cdot r^{-\alpha}}{N_0 + P\cdot\beta\cdot R^{-\alpha}}$. □

Lemma 5: $\mathbb{S}_{i,j}$ is a CTCS when $R \ge (c_9\cdot r^{-\alpha} + c_{10})^{-1/\alpha}$, where $c_9 = \frac{\eta_2}{\beta\cdot\ln\eta_1(1-\sqrt[\eta_3]{p_o})^{-1}}$ and $c_{10} = -\frac{N_0}{P\cdot\beta}$.

Proof: Please refer to [24] for the details. □

Now, we are ready to prove Theorem 1.

Proof of Theorem 1: From Lemma 5, we know that when $R \ge (c_9\cdot r^{-\alpha} + c_{10})^{-1/\alpha}$, $\mathbb{S}_{i,j}$ is a CTCS. Since large $|\mathbb{S}_{i,j}|$ implies more concurrent data transmissions, we prefer a small R. Thus, let $R = (c_9\cdot r^{-\alpha} + c_{10})^{-1/\alpha}$. Then, for a large n, *i.e.* large-scale WSNs, $R \sim \Theta(r + o(1))$. Let $\omega = \Theta(\frac{r+o(1)}{l})$, which implies $R = \omega l$. Thus, Theorem 1 holds. □

[4]This can be proven by *mathematical induction*.

From Theorem 1, we know that if we set $R = \omega \cdot l$, then, all the CTCSs can conduct data transmissions simultaneously in an interference-free manner. Based on the conclusion of Theorem 1, the following corollary can be obtained.

Corollary 1: By $\vec{X}$ and $\vec{Y}$, the cells $\kappa_{i,j}$ $(1 \leq i, j \leq m)$ can be partitioned into at most ω^2 CTCSs (equivalence classes).

IV. Snapshot Data collection

In this section, we study the achievable network capacity of snapshot data collection. First, we propose a novel Cell-based Multi-Path Scheduling (CMPS) algorithm for snapshot data collection. Subsequently, we analyze the achievable network capacity of CMPS. Finally, we make some further discussion about the extension from snapshot data collection to continuous data collection.

A. Cell-based Multi-Path Scheduling (CMPS)

Before presenting the CMPS algorithm, we construct a *data collection tree* which serves as the routing structure for the data collection process. For each cell $\kappa_{i,j}$ $(1 \leq i, j \leq m)$, we abstract it into a *super node*, denoted by $s^u_{i,j}$[5]. Following the discussion in Section III, a cell contains $2\log n$ sensors in the average case and $6\log n$ sensors in the worst case. Thus, we abstract the data packets of sensors within a cell as a *super data packet*, whose size is $2\log n \cdot B$ bits in the average case and $6\log n \cdot B$ bits in the worst case. Accordingly, to send out a super data packet, we define a *super time slot* t_s as $2\log n \cdot t_m$ in the average case and $6\log n \cdot t_m$ in the worst case. Afterwards, considering the communication modes defined in Section III, we construct a data collection tree, denoted by $\mathbb{T}$, rooted at the sink to connect all the super nodes according to the following rules:

1) For super nodes $s^u_{i,j}$ $(1 \leq i, j \leq m')$ (note that $m' = m-1$), $s^u_{i,j}$ transmits its data to $s^u_{i+1,j+1}$, creating a link from $s^u_{i,j}$ to $s^u_{i+1,j+1}$.
2) For super nodes $s^u_{m,j}$ $(1 \leq j \leq m')$, $s^u_{m,j}$ transmits its data to $s^u_{m,j+1}$, creating a link from $s^u_{m,j}$ to $s^u_{m,j+1}$.
3) For super nodes $s^u_{i,m}$ $(1 \leq i \leq m')$, $s^u_{i,m}$ transmits its data to $s^u_{i+1,m}$, creating a link from $s^u_{i,m}$ to $s^u_{i+1,m}$.

After applying the above rules to all the super nodes except for $s^u_{m,m}$, the data collection tree is built. Taking the WSN shown in Fig.1 as an example, the obtained data collection tree is shown in Fig.3. For a data transmission route from a leaf super node to the root in $\mathbb{T}$, we call it a *path*. The path starting from $s^u_{i,1}$ $(1 \leq i \leq m)$ is denoted by P_i and the path from $s^u_{1,j}$ $(2 \leq j \leq m)$ is denoted by P'_j, as shown in Fig.3.

According to Corollary 1, all the cells of a WSN can be partitioned into ω^2 CTCSs (equivalence classes). For each CTCS $\mathbb{S}_{i,j}$ $(1 \leq i, j \leq \omega)$, we map it to an integer $(i-1)\cdot\omega+j$. In Fig.3, the number next to each super node indicates the CTCS it belongs to. For convenience, we also use $\mathbb{S}_{(i-1)\cdot\omega+j}$ to represent the CTCS $\mathbb{S}_{i,j}$ $(1 \leq i, j \leq \omega)$.

Based on the abstracted data collection tree $\mathbb{T}$, we propose a novel *Cell-based Multi-Path Scheduling* (CMPS) algorithm,

[5]To minimize confusion, we use cell and super node interchangeably.

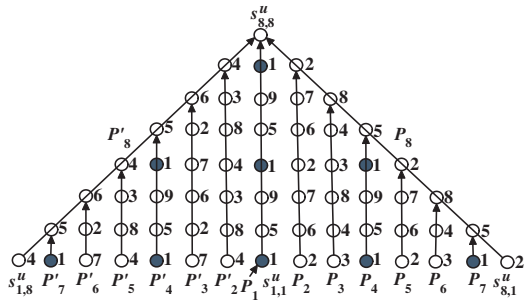

Fig. 3. A data collection tree.

which has two phases. In Phase I, we schedule the ω^2 CTCSs one by one, until all the data packets of cells $\kappa_{i,j}$ $(1 \leq i, j \leq m')$ have been collected by the cells on path P_m, path P'_m, or the sink. In Phase II, we schedule the cells of P_m and P'_m until all the data packets have been collected by the sink. We use the example shown in Fig.3 to present the CMPS procedure as follows.

Phase I: Inner-Tree Scheduling. Since the cells within a CTCS can be scheduled to transmit data concurrently, schedule CTCSs $\mathbb{S}_1, \mathbb{S}_1, \cdots, \mathbb{S}_{\omega^2}$ orderly, each for a super time slot. Repeat Phase I until there is no packet remaining at the super node $s^u_{i,j}$ $(1 \leq i, j \leq m')$, *i.e.* all the data packets at $s^u_{i,j}$ $(1 \leq i, j \leq m')$ have been collected by the sink or $s^u_{i,j}$ $(i = m$ or $j = m)$. For the specific sensors within a cell, schedule them sequentially according to any order at the available super time slots for this cell[6]. Taking the data collection tree $\mathbb{T}$ shown in Fig.3 as an example, the cells in $\mathbb{T}$ can be partitioned into 9 CTCSs. For the 9 CTCSs $\mathbb{S}_1, \mathbb{S}_1, \cdots, \mathbb{S}_9$, we schedule them sequentially each for one super time slot. At the end of Phase I, all the data packets of $s^u_{i,j}$ $(1 \leq i, j \leq 7)$ have been collected to the sink, or the cells on path P_8 and P'_8.

Phase II: Scheduling of P_m and P'_m. For the super nodes $s^u_{i,j}$ $(i = m$ or $j = m)$ which have data packets to be collected, partition them into λ CTCSs ($\lambda \leq 2\omega - 1$ which is proven in Lemma 9.). Then, schedule these λ CTCSs sequentially each for one super time slot. Repeat Phase II until all the packets have been collected by the sink. Taking P_8 and P'_8 shown in Fig.3 as an example, the cells on P_8 and P'_8 can be partitioned into 5 CTCSs. Then, we schedule these 5 CTCSs sequentially until all the data packets have been collected to the sink.

B. Capacity Analysis of CMPS

In this subsection, we investigate the achievable network capacity of CMPS. The upper bound of SDC is W even under

[6]Suppose the parent node of super node $s^u_{i,j}$ is $s^u_{i',j'}$, *i.e.* all the nodes in cell $\kappa_{i,j}$ will transmit their data to the nodes in cell $\kappa_{i',j'}$. Then, when a node s_u in cell $\kappa_{i,j}$ is scheduled to transmit data to some node in cell $\kappa_{i',j'}$, s_u will transmit its data to the node s_v in cell $\kappa_{i',j'}$, where s_v satisfies the condition that the success probability of the link from s_u to s_v is the highest among the links from s_u to all the nodes in cell $\kappa_{i',j'}$.

Now, assume the success probability of the link from s_u to s_v is 0.5. Then, when s_u transmits a data packet to s_v, s_v successfully receives this data packet with probability 0.5. If this data transmission fails, s_u will retransmit that data packet until the packet is successfully received by s_v. Evidently, the expected transmission times of that packet is 2 in this case.

In this paper, without specification, for any sensor node s_u, it determines its next hop and transmits data in terms of the aforementioned manner.

the DNM [4], [8]. Therefore, the upper bound of SDC under the PNM is also W. Consequently, we focus on the lower bound of CMPS in the following analysis.

For convenience, we introduce the concept of a *scheduling round*. A scheduling round for Phase I (Phase II) of CMPS is the time used to run Phase I (Phase II) once. For the data collection tree $\mathbb{T}$ shown in Fig.3, a scheduling round is $9t_s$ ($5t_s$) in Phase I (Phase II), since there are 9 (5) CTCSs which need to be scheduled for each run of Phase I (Phase II). Now, we can obtain the number of super time slots used in Phase I of CMPS as shown in Lemma 6.

Lemma 6: For SDC, CMPS takes $\omega^2 m'$ super time slots to finish Phase I, where $m' = m - 1$.

Proof: Please refer to [24] for details. □

Now, we study the number of time slots used in Phase II. First, we derive the remaining number of the super data packets at each of the super nodes $s^u_{i,j}$ ($i = m$ or $j = m$) to be transmitted at the beginning of Phase II. Subsequently, we obtain the upper bound of the number of super time slots used in Phase II, followed by the lower bound of the achievable network capacity of CMPS. In the following analysis, we use $\phi_{i,j}$ $(1 \le i, j \le m)$ to denote the number of the super data packets transmitted/forwarded by $s^u_{i,j}$ through the entire SDC process. Further, we use $\varphi_{i,j}$ $(1 \le i, j \le m)$ to denote the number of the super data packets at $s^u_{i,j}$ to be transmitted at the beginning of Phase II. Clearly, $\varphi_{i,j} = 0$ $(1 \le i, j \le m')$ after Phase I.

Lemma 7: For $1 \le i \le m'$, $\phi_{m,i} = \frac{i(i+1)}{2}$.

Proof: Please refer to [24] for details. □

From the proof of Lemma 7 and by symmetry, we have the following corollary.

Corollary 2: For $1 \le i \le m'$, $\phi_{i,m} = \frac{i(i+1)}{2}$.

Based on Lemma 7, we obtain the number of the super data packets at $s^u_{m,i}$ to be transmitted at the beginning of Phase II as shown in Lemma 8.

Lemma 8: Let $\theta = \lceil \frac{\sqrt{1+8m'}-1}{2} \rceil$, then

$$\varphi_{m,i} = \begin{cases} 0, & 1 \le i < \theta \\ \phi_{m,i} - m' = \frac{i(i+1)}{2} - m' \le i, & i = \theta \\ i, & \theta < i \le m' \end{cases}$$

Proof: We prove this lemma by cases.

Case 1: $1 \le i < \theta$. From Lemma 7, $s^u_{m,i}$ transmits/forwards $\phi_{m,i} = \frac{i(i+1)}{2}$ super data packets to its parent through the entire SDC process. In Phase I, we schedule every CTCS for m' times by the proof of Lemma 6, which implies $s^u_{m,i}$ has been scheduled for m' times. It follows that $s^u_{m,i}$ can transmit/forward m' super data packets to its parent during its available super time slots in Phase I. Considering that $1 \le i < \theta$, we have $\phi_{m,i} = \frac{i(i+1)}{2} \le \frac{1}{2}(\theta^2 + \theta) = \frac{1}{2} \cdot 2m' = m'$. Thus, we conclude that $s^u_{m,i}$ $(1 \le i < \theta)$ has already finished its data transmission task in Phase I, *i.e.* $\varphi_{m,i}(1 \le i < \theta) = 0$ at the beginning of Phase II.

Case 2: $i = \theta$. According to the proof of the previous case and the scheduling of Phase I, for super node $s^u_{m,\theta}$, its two children $s^u_{m,i-1}$ and $s^u_{m-1,i-1}$ have no data packet to be transmitted at the beginning of Phase II. Furthermore, as explained in the previous case, $s^u_{m,\theta}$ has been scheduled for m' times in Phase I, which implies that $s^u_{m,i}$ transmitted m' super data packets to its parent. It follows that the number of data packets to be transmitted at $s^u_{m,\theta}$ is $\varphi_{m,i} = \phi_{m,i} - m' = \frac{i(i+1)}{2} - m' \le i$ at the beginning of Phase II.

Case 3: $\theta < i \le m'$. For the child $s^u_{m-1,i-1}$ of $s^u_{m,i}$, it transmitted $i - 1$ super data packets to $s^u_{m,i}$ in Phase I by the proof of Lemma 7. For another child $s^u_{m,i-1}$ of $s^u_{m,i}$, it transmitted m' super data packets to $s^u_{m,i}$ in Phase I by the proof of Lemma 6. Furthermore, $s^u_{m,i}$ also transmitted m' super data packets to its parent $s^u_{m,i+1}$ by the proof of Lemma 6. This implies the number of super data packets to be transmitted at $s^u_{m,i}$ $(\theta < i \le m')$ at the beginning of Phase II is $\varphi_{m,i} = (i-1) + 1 = i$. □

According to Corollary 2 and Lemma 8, it is straightforward to obtain the following corollary by symmetry.

Corollary 3:

$$\varphi_{i,m} = \begin{cases} 0, & 1 \le i < \theta \\ \phi_{m,i} - m' = \frac{i(i+1)}{2} - m' \le i, & i = \theta \\ i, & \theta < i \le m' \end{cases}$$

Lemma 9: For super nodes $s^u_{m,i}$ $(\theta \le i \le m')$ and $s^u_{j,m}$ $(\theta \le j \le m')$, they can be partitioned into at most $2\omega - 1$ CTCSs, *i.e.* $\lambda \le 2\omega - 1$, where λ is the one in Phase II of CMPS.

Proof: Please refer to [24] for details. □

Lemma 10: In Phase II of the CMPS algorithm, it costs at most $\frac{1}{2}(2\omega-1)(m'+\theta)(m'-\theta+1)$ super time slots to transmit all the data packets to the sink.

Proof: Please refer to [24] for details. □

Now, we are ready to derive the achievable network capacity of CMPS in both the worst case and the average case as shown in the following theorem.

Theorem 2: For the achievable data collection capacity of CMPS for SDC, it is $\Omega(\frac{p_o}{3\omega} \cdot W)$ in the worst case and $\Omega(\frac{p_o}{\omega} \cdot W)$ in the average case. In both cases, the achievable data collection capacity of CMPS is order-optimal.

Proof: From Lemma 6 and Lemma 10, the total number of the super time slots used by CMPS is at most $\omega^2 m' + \frac{1}{2}(2\omega - 1)(m' + \theta)(m' - \theta + 1) \le \omega^2 m + \omega m^2 \le O(\frac{\omega n}{2 \log n})$. The total amount of data received by the sink is $n \cdot B$. Thus, in the worst case the achievable network capacity of CMPS is

$$\frac{n \cdot B}{O(\frac{\omega n}{2 \log n}) \cdot t_s} = \frac{n \cdot B}{O(\frac{\omega n}{2 \log n}) \cdot 6 \log n \cdot t_m} = \Omega(\frac{p_o}{3\omega} \cdot W). \tag{14}$$

Similarly, in the average case, the achievable network capacity of CMPS is

$$\frac{n \cdot B}{O(\frac{\omega n}{2 \log n}) \cdot t_s} = \frac{n \cdot B}{O(\frac{\omega n}{2 \log n}) \cdot 2 \log n \cdot \frac{t_o}{p_o}} = \Omega(\frac{p_o}{\omega} \cdot W). \tag{15}$$

Since the upper bound of SDC is W under both the deterministic and probabilistic network models, and p_o, ω are

constants, the achievable network capacity of CMPS in both cases is order-optimal. □

When addressing the CDC problem, an intuitive idea is to combine the existing SDC methods with the *pipeline* technique. Nevertheless, such an idea cannot induce a significant improvement on the network capacity. Taking the CMPS as an example, it has already achieved the order-optimal data collection capacity. By pipelining the CMPS algorithm, data transmissions at the nodes far from the sink can definitely be accelerated. However, the fact that the sink can receive at most one packet during each time slot makes the data accumulated at the nodes near the sink. As a result, the network capacity still cannot be improved even with pipeline [8].

V. Continuous Data Collection

Since the combination of a SDC method and the *pipeline* technique cannot improve the network capacity effectively, we propose a novel *Zone-based Pipeline Scheduling* (ZPS) algorithm based on *Compressive Data Gathering* (CDG) [15].

A. Zone-based Pipeline Scheduling (ZPS)

CDG is first proposed in [15] for distributing the SDC load uniformly to all the nodes in a network. Suppose P is a data collection path consisting of d nodes $s_1, s_2, \cdots, s_d$, where s_1 is the leaf node, s_d is the sink (destination), and the data (packet) produced at s_i $(1 \leq i \leq d-1)$ is D_i. We use the data collection process on P to show the basic idea of CDG. In the traditional data collection way, for node s_i $(1 \leq i \leq d-1)$ on P, it transmits i data packets to its parent (1 is for itself and $i-1$ is for the packets it received), which is unbalanced, *i.e.* the nodes near the sink transmit more data than the ones far from the sink. By contrast, to collect D_i $(1 \leq i \leq d-1)$ to the sink, every node transmits M data packets to its parent under the CDG way, *i.e.* s_1 multiplies its data with M random coefficients ψ_{i_1} $(1 \leq i \leq M)$ respectively, and sends the M new data (packets) $\psi_{i_1} D_1$ $(1 \leq i \leq M)$ to its parent s_2; after s_2 receives these M data (packets) from s_1, s_2 first multiplies its data with M random coefficients ψ_{i_2} $(1 \leq i \leq M)$ respectively, adds $\psi_{i_2} D_2$ with $\psi_{i_1} D_1$ respectively, and subsequently sends M new results $\psi_{i_1} D_1 + \psi_{i_2} D_2$ $(1 \leq i \leq M)$ to its parent s_3; for the subsequent nodes s_i $(3 \leq i \leq d-1)$, it does the similar multiplication-addition operations as s_2, and sends the M new results $\sum_{j=1}^{i} \psi_{1_j} D_j, \sum_{j=1}^{i} \psi_{2_j} D_j, \cdots, \sum_{j=1}^{i} \psi_{M_j} D_j$ to its parent s_{i+1}. Finally, after s_d receives all the M packets $\sum_{j=1}^{d-1} \psi_{1_j} D_j, \sum_{j=1}^{d-1} \psi_{2_j} D_j, \cdots, \sum_{j=1}^{d-1} \psi_{M_j} D_j$ from s_{d-1}, it can restore the original D_i $(1 \leq i \leq d-1)$ based on the compressive sampling theory [15]. For the parameter M in CDG, usually $M \ll n$ for large-scale WSNs.

Considering the benefit brought by CDG, we combine it with the pipeline technique to design an efficient CDC algorithm, named the *Zone-based Pipeline Scheduling* (ZPS) algorithm. Before presenting the detailed design of ZPS, we further partition the data collection tree $\mathbb{T}$ constructed in Section IV-A into *levels* and *segments*, which are sets of cells (super nodes) and compatible zones, respectively. As shown in Section III, a WSN can be partitioned into $(\lceil m/\omega \rceil)^2$ compatible zones. For these zones, we define the set $\{o_{j,i}, o_{i,j} | i \leq j \leq \lceil m/\omega \rceil\}(1 \leq i \leq \lceil m/\omega \rceil)$ as a *segment*, denoted by S_i $(1 \leq i \leq \lceil m/\omega \rceil)$. Within segment S_i $(1 \leq i \leq \lceil m/\omega \rceil)$, we define the set $\{s^u_{y,x}, s^u_{x,y} | x = (i-1) \cdot \omega + j, x \leq y \leq m\}$ $(1 \leq j \leq \omega)$ as a *level*, denoted by L^i_j $(1 \leq j \leq \omega)$. Taking the $\mathbb{T}$ shown in Fig.3 as an example, it can be partitioned into 3 segments. Within a segment, the super nodes can be partitioned into 3 levels, *e.g.* within S_2, $L^2_3 = \{s^u_{6,6}, s^u_{7,6}, s^u_{8,6}, s^u_{6,7}, s^u_{6,8}\}$.

Based on the definitions of segments, levels and CTCSs, we observe that (i) for levels L^i_j $(1 \leq i \leq \lceil m/\omega \rceil)$, all their super nodes (cells), *i.e.* $\bigcup_{i=1}^{\lceil m/\omega \rceil} L^i_j$, come from CTCSs $\mathbb{S}_{j,k} \bigcup \mathbb{S}_{j,k} (j \leq k \leq \omega)$, *i.e.* the super nodes in $\bigcup_{i=1}^{\lceil m/\omega \rceil} L^i_j$ can be partitioned into at most $2\omega - 1$ CTCSs; and (ii) on the other hand, for every super node (cell) in $\mathbb{S}_{j,k} \bigcup \mathbb{S}_{j,k} (j \leq k \leq \omega)$, it is located at level L^i_j for some $1 \leq i \leq \lceil m/\omega \rceil$. According to the observations, we design a *Zone-based Pipeline Scheduling* (ZPS) algorithm for CDC, which consists of *inter-segment pipeline scheduling* and *intra-segment scheduling* as follows.

Inter-Segment Pipeline Scheduling. Since the super nodes at levels L^i_j $(1 \leq i \leq \lceil m/\omega \rceil)$ can be partitioned into $2\omega - 1$ CTCSs, we can take each level as a unit and schedule the j-th $(1 \leq j \leq \omega)$ level of all the segments S_i $(1 \leq i \leq \lceil m/\omega \rceil)$ simultaneously. In other words, we can schedule all the segments concurrently as long as we schedule the same j-th $(1 \leq j \leq \omega)$ level within each segment. Therefore, when collecting N continuous snapshots, we can pipeline the data transmission on the segments, *i.e.* for each segment S_i $(1 \leq i \leq \lceil m/\omega \rceil)$, S_i starts to transmit the data packets of the $(k+1)$-th $(k > 0)$ snapshot immediately after it transmits all the data of the k-th snapshot to segment S_{i+1}. Suppose $t(S_i)(1 \leq i \leq \lceil m/\omega \rceil)$ is the number of the super time slots used by segment S_i to transmit all the data packets of a snapshot to the subsequent segment (or the sink) and let $t_p = \max\{t(S_i) | 1 \leq i \leq \lceil m/\omega \rceil\}$. Then, a segment data transmission pipeline on all the segments is formed with each segment working with t_p super time slots for every snapshot (Now, a snapshot is equivalent to an individual task in a traditional pipeline operation). By this data transmission pipeline, the sink can receive the data of a snapshot in every t_p super time slots after it receives the data of the first snapshot.

Intra-Segment Scheduling. Within segment $S_i (1 \leq i \leq \lceil m/\omega \rceil)$, to transmit the k-th snapshot, we schedule the super nodes level by level, *i.e.* schedule $L^i_1, L^i_2, \cdots, L^i_\omega$ sequentially to transmit the k-th snapshot. Finally, the data packets of the kth snapshot are transmitted to the next segment by the super nodes at level L^i_ω. When schedule L^i_j $(1 \leq j \leq \omega)$ for the k-th snapshot, we first partition the super nodes in L^i_j into at most $2\omega - 1$ CTCSs according to the observations. Subsequently, we schedule these $2\omega - 1$ CTCSs sequentially. When schedule a particular CTCS, we let all the super nodes within this CTCS transmit their data in the CDG way, *i.e.* for every super node in this CTCS, it first does the similar multiplication-addition operations as in CDG, and then transmits the M new obtained results to its parent at the subsequent level. Thus, to schedule

a CTCS in the CDG way takes M super time slots instead of one. However, this way is more suitable for the pipeline operation by avoiding data accumulation at the sensors near the sink.

B. Capacity Analysis of ZPS

In this subsection, we analyze the achievable data collection capacity of ZPS to collect N continuous snapshots. First, we investigate the consumed time slots to collect the first snapshot, which is the foundation of the data collection pipeline. Subsequently, we derive the achievable CDC capacity of ZPS for different cases.

Lemma 11: (i) For the t_p in the inter-segment pipeline scheduling of ZPS, $t_p \leq \omega(2\omega-1)M$; (ii) The number of the super time slots used to collect the first snapshot is at most $\lceil \frac{m}{\omega} \rceil \omega(2\omega-1)M$.

Proof: Please refer to [24] for details. □

Theorem 3: To collect N continuous snapshots, the achievable network capacity of ZPS is

$$\begin{cases} \Omega(\frac{p_o N\sqrt{n/\log n}}{6\sqrt{2}\omega M} \cdot W + o(1)), N = O(\sqrt{n/\log n}) \\ \Omega(\frac{p_o n}{12\omega^2 M \log n} \cdot W + o(1)), N = \Omega(\sqrt{n/\log n}) \end{cases} \quad (16)$$

in the worst case. The achievable network capacity of ZPS is

$$\begin{cases} \Omega(\frac{p_o N\sqrt{n/\log n}}{2\sqrt{2}\omega M} \cdot W + o(1)), N = O(\sqrt{n/\log n}) \\ \Omega(\frac{p_o n}{4\omega^2 M \log n} \cdot W + o(1)), N = \Omega(\sqrt{n/\log n}) \end{cases} \quad (17)$$

in the average case.

Proof: To collect N continuous snapshots, the data transmission process can be pipelined according to ZPS, which implies the sink can receive the data of a snapshot every t_p super time slots after it receives the first snapshot. Therefore, by Lemma 11, the number of the super time slots used to collect N continuous snapshots is at most $\lceil \frac{m}{\omega} \rceil \omega(2\omega-1)M + (N-1)\omega(2\omega-1)M \leq (\frac{m}{\omega}+1) \cdot 2\omega^2 M + 2\omega^2(N-1)M = O(2\omega mM + 2\omega^2 NM)$. Thus, in the worst case, the achievable network capacity of ZPS is at least $\frac{NnB}{O(2\omega mM+2\omega^2 NM)\cdot 6\log n \cdot t_m}$ = Eq. 16. In the average case, the achievable network capacity of ZPS is at least $\frac{NnB}{O(2\omega mM+2\omega^2 NM)\cdot 2\log n \cdot t_m}$ = Eq. 17. □

From Theorem 3, we know that the achievable network capacity of ZPS is $\sqrt{\frac{n}{\log n}}$ or $\frac{n}{\log n}$ times better than the optimal capacity in order, which is a very significant improvement. By examining ZPS carefully, we find the *pipeline scheduling* and CDG are responsible for this improvement. By forming a CDG based pipeline, the time overlap of gathering multiple continuous snapshots conserves a lot of time, which accelerates the data collection process directly and significantly.

VI. Simulations and Result Analysis

We validate the effectiveness of the proposed algorithms via simulations. For all the simulations, we consider a WSN with one sink where all the senor nodes are randomly distributed in a square area. The time is slotted, and the size of each time slot is normalized to one. Every node produces one data packet in a snapshot and the size of a packet is normalized to one. All the nodes work with the same initial power P over a common wireless channel whose bandwidth is also normalized to one. Further, we define the node density of a WSN as ρ. In all the following simulations, we set $\rho = 3$. For the other parameters, we set them by referring the settings in [17] and they are given in Table I, where the parameters have the same meanings as in previous sections. Moreover, each group of simulations is repeated for 100 times and the results are the average values.

TABLE I
Parameters.

Para.	Value
α	3.0
η_1	.25
η_2	10.0
η_3	10.0
P/N_0	10.0
ρ	3.0
M	50
N	1000

TABLE II
SDC capacity.

p_o	PS	MPS	CMPS
.60	.165	.2018	.2377
.65	.1729	.2172	.249
.70	.1801	.2235	.2589
.75	.1864	.2394	.2688
.80	.1913	.2466	.2759
.85	.1953	.2596	.2813
.90	.1967	.2611	.283
.95	.1934	.2574	.2785

Since there is no existing data collection algorithms for probabilistic WSNs currently, we compare our proposed algorithms with the latest data collection algorithms for deterministic WSNs. We compare CMPS with PS [4] and MPS [8], [9], and compare ZPS with PSA [8], [9] and CDG [15] (see Section V). PS (*Breadth First Searching tree* (BFS)-based) and MPS (*Connected Dominating Set* (CDS)-based) are the latest data collection algorithms for SDC under the DNM. PSA is the latest continuous data collection algorithm proposed by us under the PrIM for deterministic WSNs, which is also involving pipeline and CDG. For fairness, we also apply the pipeline technique in CDG when comparing it with ZPS.

A. Performance of CMPS

We implement PS, MPS, and CMPS in a probabilistic WSN deployed in an area of 100×100 for SDC, and the achievable capacities are shown in Table II for different p_o values. From Table II we know that when p_o varies from 0.6 to 0.9, the capacities of PS, MPS, and CMPS increase. This is because a higher p_o implies fewer average transmission times over a lossy link (note that the average number of transmission times over a lossy link is $\frac{1}{p_o}$). Consequently, with the increasing of p_o, the capacities of PS, MPS, and CMPS increase. However, when p_o varies from 0.9 to 0.95, the capacities of PS, MPS, and CMPS decrease. This is because, on the other hand, a higher p_o also implies a larger R from the proof of Lemma 5. Whereas, larger R implies fewer parallel transmissions can be concurrently conducted, which leads to the decrease of the achievable capacities of PS, MPS, and CMPS. Note that although the capacities of CMPS for the case $p_o = 0.8$ and the case $p_o = 0.95$ are similar, they have quite different meanings. Since small p_o implies more average transmission times, a network consumes less energy in the case $p_o = 0.95$ than that in the case $p_o = 0.8$ even they have similar capacities.

From Table II we can also see that CMPS always achieves a higher network capacity compared with PS and MPS. This is because that PS is a single path scheduling algorithm performed on a BFS tree. While CMPS schedules a CTCS each super time slot, which is equivalent to schedule multiple cells

TABLE III
CDC CAPACITY.

p_o	100×100			200×200		
	PSA	CDG	ZPS	PSA	CDG	ZPS
.60	.7107	.5963	.8939	2.9094	2.131	3.4154
.65	.7158	.6098	.9059	2.9216	2.1457	3.4416
.70	.7169	.615	.9062	2.9373	2.1513	3.4465
.75	.7114	.6121	.9009	2.9204	2.137	3.4295
.80	.6925	.6013	.8868	2.8103	2.1038	3.3717
.85	.6686	.5842	.8631	2.7193	2.0442	3.276
.90	.6218	.5554	.819	2.5201	1.9413	3.1102
.95	.5574	.5014	.7463	2.2235	1.7656	2.8391

on multiple paths. In other words, CMPS achieves complete concurrency by scheduling multiple cells. Furthermore, the CDS-based data collection tree used by MPS is unbalanced and does not consider lossy links, which leads to the degradation of its capacity.

B. Performance of ZPS

To compare the performances of ZPS, PSA, and CDG for CDC, we conduct two groups of simulations in probabilistic WSNs of sizes 100×100 and 200×200 for different p_o values, respectively, as shown in Table III. From Table III we can see that, due to the same reasons discussed before, when p_o varies from 0.6 to 0.7, the capacities of PSA, CDG, and ZPS increase. By contrast, when p_o varies from 0.75 to 0.95, the capacities of PSA, CDG, and ZPS decrease. Additionally, the performance of all these three algorithms highly depends on the data transmission pipeline, larger p_o (*i.e.* larger R) implies larger segments (*i.e.* larger t_p in Section V), which further leads to the decrease of the capacities of PSA, CDG, and ZPS.

We can also see from Table III that with the increase of the network size (*i.e.* the number of sensors in a WSN), the achievable capacities of all the algorithms increase. This is because that the data transmission pipeline is easier to form and more effective in large WSNs. On the other hand, since all these three algorithms addressed the data accumulation problem, they can form effective data transmission pipelines, which finally make them achieve higher capacities. Furthermore, ZPS achieves a higher capacity than PSA and CDG. This is due to (i) when constructing data collection trees, PSA and CDG do not consider lossy links; (ii) the data collection trees used by PSA and CDG may be very unbalanced, which obstructs to form effective data transmission pipelines. By contrast, the data collection tree used by ZPS is balanced and has a more reasonable structure, which is more suitable to form a pipeline; (iii) ZPS has a more sound scheduling scheme compared with CDG, *i.e.* a WSN is partitioned into multiple CTCSs, and ZPS achieves complete concurrency while scheduling these CTCSs.

VII. CONCLUSION

Currently, there exist no works that study the data collection capacity issue for probabilistic WSNs under the practical PNM. The work proposed in this paper fills this gap. For SDC, we propose a novel Cell-based Multi-Path Scheduling (CMPS) algorithm, which schedules multiple super nodes on multiple paths concurrently. Theoretical analysis shows that CMPS has achieved the order-optimal network capacity. For CDC, we propose a Zone-based Pipeline Scheduling (ZPS) algorithm. ZPS significantly speeds up the CDC process by forming a data transmission pipeline, and achieves a surprising network capacity of $\sqrt{n/\log n}$ or $n/\log n$ times better than the optimal capacity in order.

ACKNOWLEDGEMENT

This work is partly supported by the NSF under grants No. CNS-1152001 and No. CNS-1052769.

REFERENCES

[1] M. Andrews and M. Dinitz. Maximizing Capacity in Arbitrary Wireless Networks in the SINR Model: Complexity and Game Theory. Infocom 2009.
[2] V. Bhandari and N. H. Vaidya. Connectivity and Capacity of Multi-Channel Wireless Networks with Channel Switching Constraints. Infocom 2007.
[3] C.-K. Chau, M. Chen, and S. C. Liew. Capacity of Large-Scale CSMA wireless Networks. MobiCom 2009.
[4] S. Chen, S. Tang, M. Huang, and Y. Wang. Capacity of Data Collection in Arbitrary Wireless Sensor Networks. Infocom 2010.
[5] O. Goussevskaia, R. Wattenhofer, M. M. Halldorsson, and E. Welzl. Capacity of Arbitrary Wireless Networks. Infocom 2009.
[6] M. Guo, X. Wang, and M. Wu. On the capacity of k-mpr wireless networks. IEEE Transactions on Wireless Communications, 8(7):3878-3886, July 2009.
[7] P. Gupta and P. R. Kumar. The capacity of wireless networks. IEEE Trans. Inform. Theo., 2000.
[8] S. Ji, Y. Li, and X. Jia. Capacity of Dual-Radio Multi-Channel Wireless Sensor Networks for Continuous Data Collection. Infocom 2011.
[9] S. Ji, Z. Cai, Y. Li, and X. Jia, Continuous Data Collection Capacity of Dual-Radio Multi-Channel Wireless Sensor Networks, IEEE Transactions on Parallel and Distributed Systems (TPDS), 2011.
[10] C. Joo, J.-G. Choi, and N. B. Shroff. Delay Performance of Scheduling with Data Aggregation in Wireless Sensor Networks. Infocom 2010.
[11] S. Li, Y. Liu, and X.-Y. Li. Capacity of Large Scale Wireless Networks Under Gaussian Channel Model. Mobicom 2008.
[12] X.-Y. Li, S.-J. Tang, and O. Frieder. Multicast Capacity for Large Scale Wireless Ad Hoc Networks. Mobicom 2007.
[13] X.-Y. Li, J. Zhao, Y. W. Wu, S. J. Tang, X. Xu, and X. F. Mao. Broadcast Capacity for Wireless Ad Hoc Networks. MASS 2008.
[14] Y. Liu, Q. Zhang, and L. M. Ni. Opportunity-based topology control in wireless sensor networks. IEEE Transactions on Parallel and Distributed Systems, 21(3):405-416, March 2010.
[15] C. Luo, F. Wu, J. Sun, and C. W. Chen. Compressive Data Gathering for Large-Scale Wireless Sensor Networks. MobiCom 2009.
[16] U. Niesen, P. Gupta, and D. Shah. The balanced unicast and multicast capacity regions of large wireless networks. IEEE Transactions on Information Theory, 56(5):2249-2271, 2010.
[17] Scott C.-H. Huang, H.-C. Wu, and P.-J. Wan. Analysis and design of a novel randomized broadcast algorithm for scalable wireless networks in the interference channels. IEEE Transactions on Wireless Communications, 9(7):2206-2215, July 2010.
[18] G. Sharma, R. Mazumdar, and N. B. Shroff. Delay and capacity trade-offs in mobile ad hoc networks: A global perspective. IEEE/ACM Transactions on Networking, 15(5):981-992, 2007.
[19] Z. Wang, H. R. Sadjadpour, and J. J. Garcia-Luna-Aceves. The Capacity and Energy Efficiency of Wireless Ad Hoc Networks with Multi-Packet Reception. Mobihoc 2008.
[20] Z. Wang, H. R. Sadjadpour, and J. J. Garcia-Luna-Aceves. A Unifying Perspective on the Capacity of Wireless Ad Hoc Networks. Infocom 2008.
[21] Y. Xu and W. Wang. Scheduling Partition for order optimal capacity in large-scale wireless networks. MobiCom 2009.
[22] M. Zuniga and B. Krishnamachari. Analyzing the Transitional Region in Low Power Wireless Links. Secon 2004.
[23] S. Ji, R. Beyah, and Y. Li, Continuous Data Collection Capacity of Wireless Sensor Networks Under Physical Interference Model, IEEE MASS 2011.
[24] S. Ji, R. Beyah, and Z. Cai, Technical Report: Snapshot/Continuous Data Collection Capacity for Large-Scale Probabilistic Wireless Sensor Networks, http://www.cs.gsu.edu/∼sji/Paper/TechReport-probabilistic.pdf.