Cell-Based Snapshot and Continuous Data Collection in Wireless Sensor Networks

SHOULING JI, Georgia State University
JING (SELENA) HE, Kennesaw State University
A. SELCUK ULUAGAC and RAHEEM BEYAH, Georgia Institute of Technology
YINGSHU LI, Georgia State University and Harbin Institute of Technology

Data collection is a common operation of wireless sensor networks (WSNs). The performance of data collection can be measured by its achievable network capacity. However, most existing works focus on the network capacity of unicast, multicast or/and broadcast. In this article, we study the snapshot/continuous data collection (SDC/CDC) problem under the physical interference model for randomly deployed dense WSNs. For SDC, we propose a *Cell-Based Path Scheduling* (CBPS) algorithm based on network partitioning. Theoretical analysis shows that its achievable network capacity is order-optimal. For CDC, a novel *Segment-Based Pipeline Scheduling* (SBPS) algorithm is proposed which combines the pipeline technique and the compressive data gathering technique. Theoretical analysis shows that SBPS significantly speeds up the CDC process and achieves a high network capacity.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Wireless sensor networks, snapshot data collection, continuous data collection

ACM Reference Format:
Ji, S., He, J. (S.), Uluagac, A. S., Beyah, R., and Li, Y. 2013. Cell-based snapshot and continuous data collection in wireless sensor networks. ACM Trans. Sensor Netw. 9, 4, Article 47 (July 2013), 29 pages.
DOI: http://dx.doi.org/10.1145/2489253.2489264

1. INTRODUCTION

Wireless sensor networks (WSNs) are mainly used for collecting data from the physical world. Data gathering can be categorized as *data aggregation* [Wan et al. 2009; Ji et al. 2012a; Yan et al. 2011; Wang et al. 2011b] and *data collection* [Chen et al. 2010, 2009b; Luo et al. 2009]. Data aggregation acquires aggregated values from WSNs, for example, maximum, minimum, or/and average values of all data, while data collection gathers all the data from a network without any data aggregation. For data collection,

This work is partly supported by the NSF under grant No. CCF-054566 and the Kennesaw State University College of Science and Mathematics Faculty Summer Research Award Program.
S. Ji is currently affiliated with Georgia Institute of Technology.
Authors' addresses: S. Ji, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0765; email: sji@gatech.edu; J. (S.) He, Department of Computer Science, Kennesaw State University, Kennesaw, GA 30144-5591; email: jhe4@kennesaw.edu; A. S. Uluagac and R. Beyah, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332; emails: {selcuk, rbeyah}@ece.gatech.edu; Y. Li (corresponding author), Department of Computer Science, Georgia State University. Atlanta, GA 30303, and School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China; email: yli@cs.gsu.edu.

DOI: http://dx.doi.org/10.1145/2489253.2489264

the union of all the sensing values from all the sensors at a particular time instance is called a *snapshot* [Chen et al. 2010]. The act of collecting one snapshot is called *snapshot data collection* (SDC). The act of collecting multiple continuous snapshots is called *continuous data collection* (CDC). To evaluate the network performance, *network capacity*, which can reflect the achievable data transmission rate, is usually used [Chen et al. 2010; Xu and Wang 2009]. For unicast, multicast, and broadcast, we use *unicast capacity*, *multicast capacity*, and *broadcast capacity* to denote the network capacity, respectively. Particularly, we use the data receiving rate at the sink, referred to as *data collection capacity*, to measure its achievable network capacity, that is, data collection capacity reflects how fast data is collected by the sink.

After Gupta and Kumar's [2000] ground breaking work in this area, many works emerged to study the network capacity issue for a variety of network scenarios, for example, multicast capacity [Li et al. 2007, 2010], unicast capacity [Niesen et al. 2010], broadcast capacity [Keshavarz-Haddad et al. 2006; Li et al. 2008; Sirkeci-Mergen and Gastpar 2007], SDC capacity [Chen et al. 2010, 2009b, Luo et al. 2009]. In the literature, three interference models are widely used; the Protocol Interference Model (PrIM) [Chen et al. 2010; Zhang et al. 2010], the Physical Interference Model (PhIM) [Chen et al. 2009a; Wang et al. 2011a], and the Generalized Physical Interference Model (GPhIM) [Li et al. 2010; Ji and Cai 2012; Li and Fang 2009; Wang et al. 2012]. Under PrIM, two nodes can successfully communicate if and only if the receiver is within the transmission range of the transmitter and out of the interference range of other simultaneous transmitters. Under PhIM, a receiver can successfully receive data from the transmitter if and only if the signal-to-interference-and-noise-ratio (SINR) of the transmitter at the receiver is no less than a threshold (denoted by η). Differently from PrIM and PhIM, where a data transmission is described by a discrete binary function, GPhIM characterizes a data transmission by a continuous data transmission-receiving function, that is, the amount of data received by the receiver from the transmitter is a continuous function depending on the SINR value at the receiver associated with the transmitter. PrIM simplifies the communication model of wireless networks by considering only local wireless interference and is therefore more convenient for analysis. By contrast, PhIM and GPhIM consider the overall aggregated wireless interference from the network for a communication. Consequently, PhIM and GPhIM are more accurate and reliable models compared with PrIM. However, many optimization problems under PhIM and GPhIM become nonconvex or even NP-hard problems, which implies that the algorithm design for wireless networks under these two models are more complicated. In this article, we consider the achievable data collection capacity for WSNs under PhIM.

Most of the existing works, studied the multicast capacity [Li et al. 2007, 2010], the unicast capacity [Niesen et al. 2010], and/or the broadcast capacity [Li et al. 2008] of wireless networks. In contrast, we study the SDC capacity and the CDC capacity of large-scale WSNs. Mulitcast/unicast/broadcast and data collection are different communication modes. Furthermore, the data collection communication mode introduces more communication traffic and wireless interference. To the best of our knowledge, only Chen et al. [2009b] considered CDC. In that work, the authors proposed a method that combines SDC scheduling and pipeline technology to carry out CDC. However, the authors also proved that their CDC method could not achieve a better network capacity compared with their SDC method, that is, both methods have the same capacity order. To fill this gap and study how to significantly improve CDC capacity, we investigate the restriction that limits pipeline technology in order to achieve a higher CDC capacity. By combining the compressive data gathering (CDG) technique [Luo et al. 2009] and the pipeline technology, we propose a new transmission and scheduling method for CDC which achieves a surprising network capacity. Particularly, the main contributions of this work are as follows.

—For a WSN deployed in a square area, we first partition the network into small *cells* and then abstract every cell as a supernode in a data collection tree rooted at the sink. Based on the data collection tree, we propose a scheduling algorithm, called *Cell-Based Path Scheduling* (CBPS), for the SDC problem in WSNs. Theoretical analysis of CBPS shows that the achievable network capacity is $\Omega(\frac{W}{4\omega})$[1], where W is the data transmission rate over a wireless channel, that is, the channel bandwidth, and ω is a specific constant depending on the wireless interference model. Since the upper bound of SDC capacity is shown to be W, CBPS successfully achieves the order-optimal network capacity.
—We propose a novel *Segment-Based Pipeline Scheduling* (SBPS) method for CDC in WSNs. SBPS combines the CDG [Luo et al. 2009] technology and the pipeline technology, which can significantly improve achievable network capacity. We theoretically prove that the asymptotic achievable network capacity of SBPS for collecting N continuous snapshots is $\Omega(\sqrt{\frac{n}{\log n}}W)$ when $N \leq \sqrt{\frac{n}{\log n}}$, or $\Omega(\frac{n}{\log n}W)$ when $N > \sqrt{\frac{n}{\log n}}$, where n is the number of sensors in a WSN. Since the current best result is $\Omega(W)$ [Chen et al. 2010, 2009b], our result is at least $\sqrt{\frac{n}{\log n}}$ or $\frac{n}{\log n}$ times better than the best one, which is a very significant improvement.
—We also conduce extensive simulations to examine the performances of the proposed algorithms. The simulation results validate that the proposed algorithms significantly improve network capacity compared with the existing works. Particularly, on average, CBPS achieves 36.9% more capacity than PS [Chen et al. 2010] for SDC, and SBPS achieves a capacity 9.39 times of that of PS [Chen et al. 2010], and 47.8% greater capacity than CDG [Luo et al. 2009] for CDC.

The rest of this article is organized as follows: Section 2 summarizes the related works and discusses the differences between this work and existing works. Section 3 introduces the network model and the network partition strategy, which is crucial for the proposed data collection methods. The Cell-Based Path Scheduling (CBPS) algorithm for snapshot data collection in WSNs is proposed and analyzed in Section 4. Section 5 presents a novel Segment-Based Pipeline Scheduling (SBPS) method for continuous data collection, and its theoretical achievable asymptotic network capacity is shown. Section 6 conducts extensive simulations for the proposed algorithms. We conclude this article in Section 7.

2. RELATED WORKS

Following the Gupta and Kumar's [2000] ground-breaking work, many subsequent works have emerged to study the network capacity issue. In this section, we summarize the existing works and discuss the differences between the existing works and ours.

2.1. Network Capacity under PrIM

[Chen et al. 2010, 2009a; Zhu et al. 2005; Ji et al. 2011; 2012b]. The upper and lower bounds of data collection capacity have been derived under PrIM for arbitrary WSNs. Li et al. [2007] investigated the multicast capacity for large-scale wireless ad hoc networks. They showed that the network multicast capacity is $\Theta(\sqrt{\frac{n}{\log n}} \cdot \frac{W}{\sqrt{k}})$ when $k = O(\frac{n}{\log n})$, and $\Theta(W)$ when $k = \Omega(\frac{n}{\log n})$, where n is the number of wireless nodes in the network, W is the bandwidth of the common channel available for multicast operations,

[1]In this article, for two functions $f(n)$ and $g(n)$, we denote $f(n) = O(g(n))$ if there is a positive constant ε such that $f(n) \leq \varepsilon g(n)$ for large enough n; $f(n) = \Omega(g(n))$ if there is a constant ε such that $f(n) \geq \varepsilon g(n)$ for large enough n.

and $k-1$ is the number of multicast sessions rooted at each node. A more general (n, m, k)-casting capacity problem under the PrIM was studied in Wang et al. [2008a], where n, m, and k denote the total number of the nodes in the network, the number of destinations of each communication group, and the actual number of communication-group members that receive information, respectively. Wang et al. [2008a], the upper and lower bounds for the (n, m, k)-casting capacity for random wireless networks.

A general framework for characterizing the network capacity of wireless ad hoc networks with arbitrary mobility patterns was studied in Garetto et al. [2007]. By relaxing the *homogeneous mixing* assumption in most existing works, the network capacity of a heterogeneous network was analyzed. Sharma et al. [2007] studied the relationship between the network capacity and the delay of mobile wireless ad hoc networks. They derived how much delay must be tolerated under a certain mobile pattern in order to achieve an improvement of the network capacity. In another similar work, Huang et al. [2010], investigated network capacity scaling in mobile wireless ad hoc networks under PrIM with infrastructure support.

Bhandari and Vaidya [2007] studied the connectivity and network capacity problems of multichannel wireless networks under PrIM. They considered a multichannel wireless network with constraints on channel switching, proposed some routing and channel assignment strategies for multiple unicast communications, and derived the per-flow capacity. Dai et al. [2008] first proposed a multichannel network architecture, called MC-MDA, and then obtained the capacity of multiple unicast communications under PrIM for arbitrary and random wireless networks. In a similar work, Zhang et al. [2010], studied the network capacity of hybrid wireless networks with directional antenna and delay constraints. Unlike previous works, Kumar et al. [2005] studied the capacity of multi-unicast for wireless networks from the algorithmic aspects, and they designed provably good algorithms for arbitrary instances. The broadcast capacity of wireless networks under PrIM is investigated in Keshavarz-Haddad et al. [2006], where the upper and lower bounds of the broadcast capacity in arbitrary connected networks were derived.

When Liu et al. [2008] studied the data gathering capacity of wireless networks under PrIM, they were concerned with the per-source node throughput in a network, where a subset of nodes send data to some designated destinations, while other nodes serve as relays. To gather data from WSNs, a multiquery processing technology is proposed [Chipara et al. 2006], considering how to obtain data efficiently with data aggregation and query scheduling. Under different communication organizations, Duarte-Melo and Liu [2003] derived the many-to-one capacity bound under PrIM. Another work studied the many-to-one capacity issue for WSNs [Marco et al. 2003], using data compression to improve data gathering efficiency. It also studied the relation between a data compression scheme and the data gathering quality. Li [2009], derived asymptotic upper bounds and lower bounds on multicast capacity for random wireless networks. Under PrIM, Liu showed that the total multicast capacity is $\Theta(\sqrt{n/\log n} \cdot (W/\sqrt{k}))$ when $k = O(n/\log n)$, and $\Theta(W)$ when $k = \Omega(n/\log n)$, where n is the number of wireless nodes, W is the bandwidth of a common wireless channel, and k is the number of nodes involving a multicast session.

The impact of the number of channels, the number of interfaces, and the interface delay on the capacity of multichannel wireless networks is investigated in Kyasanur and Vaida [2005]. In this work, the authors derived the network capacity under different situations for arbitrary and random networks. Chafekar et al. [2008] investigated the network capacity with random-access scheduling. In this work, each link is assigned a channel access probability, based on which some simple and distributed channel-access strategies are proposed.

2.2. Network Capacity under PhIM and GPhIM

Chen et al. [2009b] investigated the data collection capacity for WSNs under PhIM. They proposed a grid partition method which divides the network into small grids to collect data and then derived the network capacity. Ji and Cai [2012] studied the snapshot data collection issue for distributed WSNs. By deriving a proper carrier-sensing range for each sensor node, they designed an asynchronous distributed snapshot data collection method with a connected dominating set (CDS)-based data collection tree serving as the routing infrastructure. In another work, Ji et al. [2012c] investigated the data collection issue for probabilistic WSNs, where the impact of the existence of lossy links on the achievable data collection capacity is considered. By analysis, they obtained the data collection capacity in both the worst-case and the average case. The data collection issue and capacity in cognitive radio networks (CRNs) are studied in Cai et al. [2012]. Unlike with traditional wireless networks, the coexistence of primary users and secondary users introduces more challengeing issues to the data collection problem in CRNs. Cai et al. [2012] proposed a snapshot data collection algorithm for CRNs followed by capacity analysis.

Goussevskaia et al. [2009] considered the scheduling problem, where all the communication requests are single-hop, and all the nodes transmit at a fixed power level. They proposed an algorithm to maximize the number of links in one time slot. Unlike Goussevskaia et al. [2009], Andrews and Dinitz [2009] considered the power-control problem. A family of approximation algorithms was presented to maximize the network capacity of arbitrary wireless networks. Li et al. [2010] showed that when $k \leq \theta_1 \frac{n}{(\log n)^{2\alpha+6}}$ and $n_s \geq \theta_2 n^{1/2+\beta}$, the capacity that each multicast session can achieve is at least $c_8 \frac{\sqrt{n}}{n_s\sqrt{k}}$, where n is the number of wireless nodes in the network, n_s is the number of multicast sessions, k is the number of destinations of each multicast session, α is the path loss exponent in PhIM and GPhIM, θ_1, θ_2, and c_8 are some special constants, and $\beta > 0$ is any positive real number. Gamal [2003] studied the scaling laws of WSNs based on an antenna-sharing idea. Tang et al. [2011] investigated, the impact of deployment size on the asymptotic capacity for wireless ad hoc networks. By analysis, the authors obtained the lower and some upper bounds on both throughput capacity and transport capacity for both unicast and multicast.

Niesen et al. [2010] studied the balanced unicast and multicast capacity of a wireless network consisting of n randomly placed nodes, and obtained the characterization of the scaling of the n^2-dimensional balanced unicast and $n2^n$-dimensional balanced multicast capacity regions under the Gaussian fading channel model. Wang et al. [2012] investigated the scaling laws on multicast capacity for random extended networks and random dense networks. By employing the percolation theory, they improved both the lower bound and the upper bound on multicast capacity. Wang et al. [2011b] studied the multicast capacity of MANETs, called motioncast. They considered the network capacity of MANETs in two particular situations, the LSRM (local-based speed-restricted) model and the GSRM (global-based speed-restricted) model. Li et al. [2008] and Sirkeci-Mergen and Gastpar [2007] studied the broadcast capacity of wireless networks, where they obtained the broadcast capacity bounds under the (general) PhIM. The multi-unicast capacity of wireless networks is studied in Franceschetti et al. [2007] via percolation theory. By applying percolation theory, the authors obtained a tighter capacity bound for arbitrary wireless networks.

Luo et al. [2009] and Xu and Wang [2009] considered both the PrIM and PhIM when they studied the network capacity for wireless networks. Luo et al. [2009] studied how to distribute the data collection task to the entire network for WSNs to achieve load balancing. In this work, all the sensors transmit the same number of data packets

during the data collection process. Chau et al. [2009] investigated the network capacity of CSMA wireless networks. They formulated the models of a series of CSMA protocols and studied the network capacity of CSMA scheduling versus TDMA scheduling. Xu and Wang [2009] proposed a scheduling partition method for large-scale wireless networks. This method decomposes a large network into many small zones, then localized scheduling algorithms, which can achieve the order optimal network capacity as a global scheduling strategy, are executed in each zone independently.

Mao et al. [2008], studied the upper and lower bounds of multicast capacity for hybrid wireless networks consisting of ordinary wireless nodes and multiple base stations connected by a high-bandwidth wired network. Considering the problem of characterizing the unicast capacity scaling in arbitrary wireless networks, Niesen et al. [2009] proposed a general cooperative communication scheme. The authors also presented a family of schemes that addresses the issues between multihop and cooperative communication when the path-loss exponent is greater than three. The worst-case capacity of data collection of a WSN is studied in Moscibroda [2007] under the PhIM and PrIM. Wang et al. [2008b] investigated the capacity and energy efficiency of wireless ad hoc networks with multipacket reception under PhIM. With the multipacket reception scheme, a tight bound of the network capacity is obtained. Furthermore, the authors showed that a trade-off can be made between increasing the transport capacity and decreasing the energy efficiency. Li and Fang [2009] studied the impacts of topology and traffic pattern on the throughput capacity of hybrid wireless networks. First, they derived the per-node capacity when the base stations are regularly placed and their transmission power is large enough for directly transmitting data to any node associated with them. Subsequently, they considered the cast in which base stations are uniformly and randomly deployed and their transmission power is the same as normal nodes. Finally, they further show that the obtained results can be extended to different power propagation models.

2.3. Remarks

The following aspects distinguish our work from existing works. First, many of the previous works studied the network capacity of wireless networks under PrIM, while our work studies the achievable network capacity of WSNs under PhIM, which implies our work is more accurate and reliable when it describes wireless interferences. Second, many of the previous works investigated the unicast capacity, multicast capacity, or/and broadcast capacity of wireless networks, which are different communication modes from data collection, especially CDC. In our work, we focus on the SDC capacity and the CDC capacity of WSNs under PhIM. Third, only a few works (e.g., [Chen et al. 2009b]) considered the CDC problem. Unfortunately, none of them achieves a better network capacity compared with SDC, even with pipeline technology. In this work, we first analyze the restriction that limits existing works in regards to pipeline technology in order to achieve a higher network capacity for CDC and then propose a new transmission and scheduling method which achieves a surprising network capacity.

3. NETWORK PARTITION

In this section, we first present the network model in consideration and define the SDC capacity and CDC capacity. Subsequently, we partition the network into *cells*. Finally, we partition these cells into *interference zones*, which are crucial for our proposed algorithms. We list the frequently used notations in this article, in Table I for convenience.

Table I. Notations Used

Notation	Description
n	the number of sensors in a WSN
s_i	the ith sensor node
r	the communication radius of a sensor node
b	the size of a data packet
W	the bandwidth of the common channel
$t = b/W$	the time duration of a time slot
α	the path-loss exponent
η	the SINR threshold
l	the length of a cell
λ	the number of cells at a row/column
$\kappa_{i,j}$	the cell with coordinates (i, j)
$R = \omega \cdot l$	the side length of an interference zone
$o_{i,j}$	the interference zone with coordinates (i, j)
$v_{i,j}$	a supernode abstracted from cell $\kappa_{i,j}$
t_s	a super time slot
p_i, p'_j	paths in the abstracted data collection tree
S_i	a segment
t_p	the maximize number of time slots used by a segment
N	the number of continuous snapshots
M	a parameter in CDG

3.1. Network Model

In this article, we consider a WSN consisting of n sensors, denoted by $s_1, s_2, \ldots, s_n$, and one sink deployed in a square with area $A = cn$, where c is a constant. Furthermore, we assume the sensors are independent and identically distributed (i.i.d.). The sink is located at the top-right corner of the square.[2] The communication radius of a sensor is r. In each time interval, every sensor generates a data packet with b bits and transmits its data to the sink in a multihop fashion over a single common wireless channel with bandwidth W bits/second, that is, the data transmission rate of the common channel is W. We further assume the network time is slotted into time slots, each with length $t = b/W$ seconds. To accurately represent the wireless interference in a WSN, we consider the data collection problem under PhIM, where a receiver s_j successfully receives the transmitted data from the sender s_i if and only if the SINR of s_i at s_j is no less than a constant $\eta > 0$, that is,

$$SINR = \frac{P_i \cdot (\|s_i - s_j\|)^{-\alpha}}{N_0 + \sum_{k \neq i} P_k (\|s_k - s_j\|)^{-\alpha}} \geq \eta,$$

where P_i is the transmission power of s_i, $\|s_i - s_j\|$ 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 P_k is the transmission power of concurrent sender $s_k (1 \leq k \leq n, k \neq i)$. In this article, we assume all the sensors have the same transmission power P when they transmit data, and therefore, we have the communication radius r of a sensor satisfying $r \leq (\frac{P}{\eta N_0})^{1/\alpha}$ (the specification of r will be given in Section 3.3).

[2]Note that when the sink is in the middle of the network, one achieves 1/4 data collection capacity of the sink in the corner.

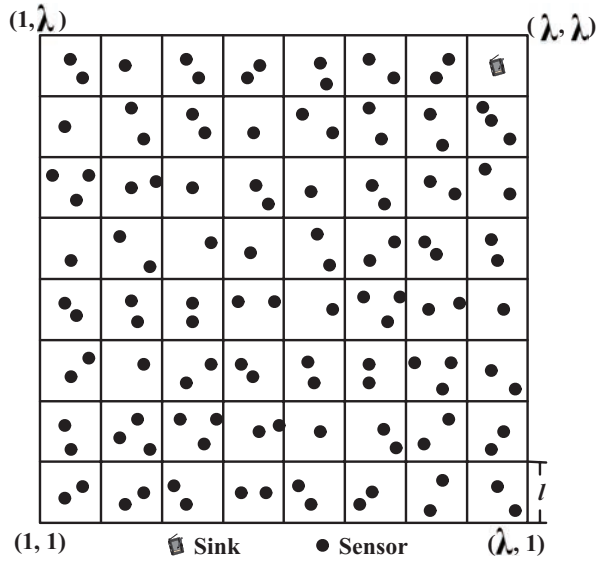


Fig. 1. Network partition.

3.2. SDC Capacity and CDC Capacity

In this article, we study the SDC and CDC problems, which are two special applications in WSNs. For SDC and CDC, only the data collected by the sink is useful. Therefore, our focus is on how fast the sink can collect the data. Consequently, considering the communication mode employed in SDC and CDC, we formally define the achievable *data collection capacity* C as the average data receiving rate of the sink during the data collection process, that is, the ratio between the amount of data successfully collected by the sink and the time τ used to collect the data. Therefore, in our WSN model, *SDC capacity* is defined as nb/τ, where nb is the amount of data in a snapshot and τ is the time consumption for collecting the data of a snapshot to the sink. Similarly, the *CDC capacity* C for collecting N continuous snapshots of data is defined by $C = Nnb/\tau$, where Nnb is the amount of data of N snapshots and τ now is the time consumption for collecting these N snapshots to the sink.

3.3. Network Partition

In the previous section, we assume the network is distributed in a square with area size $A = cn$. We partition the network into small square cells with edge length $\sqrt{2c \log n}$, denoted by l, by a group of horizontal and vertical lines. The divided network is shown in Figure 1. In order for the sensors in a cell to transmit their data to the sensors in the neighboring cells, we set a sensor's communication range $r = 2\sqrt{2}l$.[3] Furthermore, we use

$$\lambda = \sqrt{cn}/\sqrt{2c \log n} = \sqrt{n/2 \log n},$$

to denote the number of cells in each row/column and define $\lambda' = \lambda - 1$. For the cells shown in Figure 1, we assign each cell positive integer coordinates $(i, j)(1 \leq i, j \leq \lambda)$, and a cell with coordinates (i, j) is denoted by $\kappa_{i,j}$. Hence, the coordinates of the bottom-leftmost corner cell are $(1, 1)$.

[3]Consequently, our network model can be considered as the extended network model [Wang et al. 2011b].

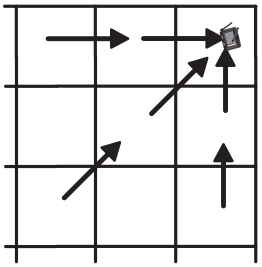


Fig. 2. Data transmission mode.

Based on this network partitioning, the following two lemmas can be derived. The proofs of these two lemmas can be found in Appendix A and Appendix B, respectively.

LEMMA 3.1. *For any cell $\kappa_{i,j}$, let e_{ij} denote the random event that cell $\kappa_{i,j}$ is empty, that is, no sensor is located at cell $\kappa_{i,j}$. Then, the probability that at least one cell is empty is no more than $\frac{1}{2n\log n}$, that is,* $\Pr[\bigcup_{1\le i,j\le\lambda} e_{ij}] \le \frac{1}{2n\log n}$.

Lemma 3.1 implies that when $n \to \infty$, the probability that one cell is empty is zero. Therefore, when n is a large value, we can assume that there is at least one sensor located in every cell.

LEMMA 3.2. *For any cell $\kappa_{i,j}$, let the random variable Z_{ij} denote the number of sensors in it. Then, the probability that cell $\kappa_{i,j}$ contains more than $8\log n$ sensors is no more than $\frac{1}{n^2}$, eight* $\Pr[Z_{ij} > 8\log n] \le \frac{1}{n^2}$.

From Lemma 3.2, the probability that a cell contains more than $8\log n$ sensors is zero when $n \to \infty$. Hence, for a large n, we use $8\log n$ as the upper bound of the number of sensors located in a cell.

3.4. Interference Zone

Since we assume the sink is located at the upper-right corner cell $\kappa_{\lambda,\lambda}$, the sensors in cell $\kappa_{i,j}$ will forward their data to the sensors located at cells $\kappa_{i+1,j}$, $\kappa_{i,j+1}$ or/and $\kappa_{i+1,j+1}$, as shown in Figure 2, that is, the sensors in each cell will forward their data to sensors in subsequent cells horizontally, vertically, or/and diagonally. Finally, the data generated by all the sensors will be forwarded to the sink via this multihop fashion.

When data transmission is initialized between two neighboring cells, they may incur interference caused by other concurrent data transmissions. To make all the concurrent data transmissions successful, we further partition the network into larger square zones with side length $R = \omega \cdot l$ (to avoid radio conflicts, $\omega > 2$, i.e. $R \ge 3l$) by another group of horizontal and vertical lines, and we call these square zones *interference zones*, as shown in Figure 3. We also assign each interference zone integer coordinates $(i, j)(1 \le i, j \le \lceil\sqrt{cn}/R\rceil)$, and interference zone (i, j) is denoted by $o_{i,j}$. For a cell $\kappa_{i',j'}$ in an interference zone $o_{i,j}$, the *relative position* of $\kappa_{i',j'}$ in $o_{i,j}$ is defined as $(i' \cdot l - (i-1)\cdot R, j' \cdot l - (j-1)\cdot R)$. We call the cells having the same relative positions in different interference zones *compatible cells*. In Figure 3, compatible cells having relative position (l, l) are highlighted. If two sensors are in different cells which are compatible cells, then they can transmit data simultaneously without incurring any interference.

At any time, we select one sensor in each compatible cell to transmit data. The selected sensors transmit data simultaneously. These transmitting sensors will not incur interference, since they are spread in different compatible cells. To this end, the

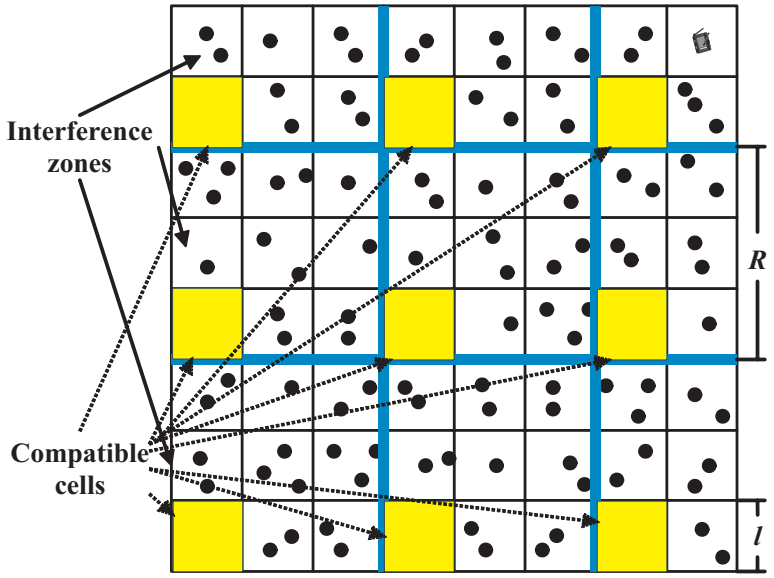


Fig. 3. Interference zones and compatible cells.

next task is to decide the value of R, as shown in Theorem 3.3.[4] The proof of Theorem 3.3 can be found in Appendix C.

THEOREM 3.3. *If we partition the network into interference zones with edge length* $R = \omega \cdot l$, *where* $\omega = 2\sqrt{2} \cdot \sqrt[\alpha]{c_1\eta}$ *is a constant value, it can be guaranteed that the sensors in compatible cells can simultaneously and successfully transmit data without interference, with each transmitting sensor residing in a unique compatible cell.*

From Theorem 3.3, the sensors from compatible cells can conduct data transmissions concurrently and successfully without interference. Therefore, by exploiting this advantage, we design SDC and CDC scheduling algorithms in the following sections which significantly accelerate the data collection process.

4. NETWORK CAPACITY OF SNAPSHOT DATA COLLECTION

In this section, we investigate the SDC problem. We propose a cell-based path scheduling algorithm based on the discussed network partition, and analyze the achievable network capacity of the proposed algorithm. Furthermore, we study the restriction that limits existing works in order to improve the network capacity, even with pipeline technology taken into account.

4.1. Cell-Based Path Scheduling (CBPS)

For cell $\kappa_{i,j}$ in a WSN, we abstract it to a *supernode*, denoted by $v_{i,j}$. Note that $v_{i,j}$ may contain at most $8 \log n$ sensors by Lemma 3.2, and we use this $v_{i,j}$ to represent the sensors in $\kappa_{i,j}$. We further define a *super time slot* $t_s = 8 \log n \cdot t$, which implies that any supernode can transmit all its data to the next hop, supernode, or sink, within a super time slot t_s. Afterwards, we construct a *data collection tree* rooted at the sink to connect

[4]Another similar conclusion is provided in Li et al. [2009] Compared with the conclusion in Li et al. [2009], we obtain a succincter and tighter result, that is, a smaller R. We also give a new proof, which is much easier to follow and understand.

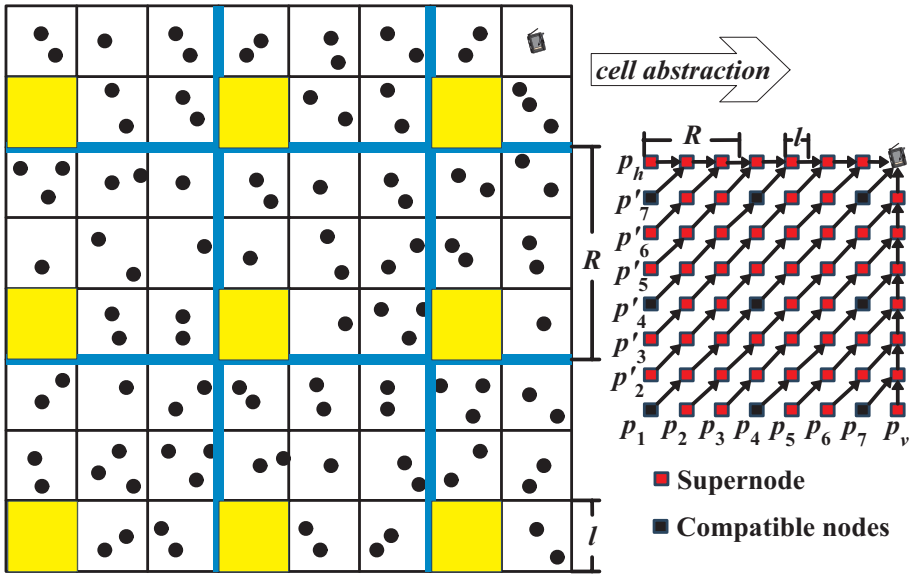


Fig. 4. Construction of a data collection tree.

all the supernodes according to the following rules. (1) For supernode $v_{\lambda',j}(1 \leq j \leq \lambda')$ (note that $\lambda' = \lambda - 1$), $v_{\lambda',j}$ transmits its data to $v_{\lambda',j+1}$, that is, creates a directed edge from $v_{\lambda',j}$ to $v_{\lambda',j+1}$; (2) for supernode $v_{i,\lambda}(1 \leq i \leq \lambda')$, $v_{i,\lambda}$ transmits its data to $v_{i+1,\lambda}$; (3) for supernode $v_{i,j}(1 \leq i, j \leq \lambda')$, $v_{i,j}$ transmits its data to $v_{i+1,j+1}$.

The abstraction and the data collection tree construction process are shown in Figure 4. In Figure 4, we also assign every path a name: p_1, the principal diagonal path; $p_i(2 \leq i \leq \lambda')$, the ith path below p_1; $p'_i(2 \leq i \leq \lambda')$, the ith path above p_1; p_v, which consists of supernodes located at the same column with the sink; and p_h, which consists of supernodes located at the same row with the sink. Furthermore, if the associated cells of some supernodes are compatible cells, then these supernodes are called *compatible nodes*. The paths that contain compatible nodes are called *compatible paths*, for example, in Figure 4, p_1, p_4, and p_7 are compatible paths. From the analysis in Section 3.4, we know that compatible nodes on compatible paths can transmit data concurrently. Then, based on compatible nodes and compatible paths, we propose a *Cell-Based Path Scheduling* (CBPS) algorithm, as shown in Algorithm 1.

The basic idea of CBPS is to partition the supernodes into different compatible supernode groups, with each group consisting of all the compatible supernodes (i.e., compatible cells). Then, these supernode groups, will be scheduled repeatedly until the data of a snapshot has been collected by the sink. We further explain our CBPS algorithm according to the example shown in Figure 4. According to Algorithm 1, CBPS has the following four steps.

Step 1 (lines 1–7). Schedule paths $p_1, p_2, \ldots, p_{\lambda'}$ until all the data packets on these paths have been transmitted to the supernodes on p_v.[5] When scheduling $p_1, p_2, \ldots, p_{\lambda'}$,

[5]Suppose $v_{i',j'}$ is the parent node of supernode $v_{i,j}$ in the data collection tree. Based on the tree construction process, $v_{i',j'}$ and $v_{i,j}$ correspond to cells $\kappa_{i',j'}$ and $\kappa_{i,j}$, respectively. Furthermore, according to Lemma 3.2, each cell contains at most $8 \log n$ sensors. In Algorithm 1, we assign each supernode a super time slot consisting of $8 \log n$ time slots every time when scheduling that supernode. Then, when scheduling supernode $v_{i,j}$, $v_{i,j}$ can transmit $8 \log n$ data packets to $v_{i',j'}$ (this also implies $v_{i,j}$ can receive at most $8 \log n$ data packets from each of its children during each of schedule). In detail, during each time of scheduling, all the sensors in cell

ALGORITHM 1: The Cell-Based Path Scheduling (CBPS) Algorithm

Require: the data collection tree, compatible cells
1: partition paths $p_1, p_2, \ldots, p_{\lambda'}$ into ω groups G_k $(0 \le k \le \omega - 1)$ for $G_k = \{p_i : 1 \le i \le \lambda', (i-1) \%\omega = k\}$
2: **for** $k = 0; i \le \omega - 1; k++$ **do**
3: partition the supernodes on the paths in G_k into ω compatible node groups g_i $(0 \le i \le \omega - 1)$
4: **for** $i = 0; i \le \omega - 1; i++$ **do**
5: schedule the supernodes in group g_i concurrently to transmit their data packets to their parent nodes
6: **end for**
7: **end for**
8: schedule paths p_i $(2' \le i \le \lambda')$ as that of the scheduling of paths p_j $(1 \le j \le \lambda')$, i.e. do lines $1-7$ for p_i $(2' \le i \le \lambda')$
9: partition the supernodes on path p_v into ω compatible node groups g_i $(0 \le i \le \omega - 1)$
10: **while** there is some data on path p_v that has not been collected by the sink **do**
11: schedule g_i $(0 \le i \le \omega - 1)$ sequentially to transmit data to their parent nodes
12: **end while**
13: schedule path p_h as that of the scheduling of p_v, i.e. do lines 9–12 for p_h

it is obvious that we can divide them into at most ω groups $G_k (0 \le k \le \omega - 1)$, with each group consisting of mutual compatible paths. Thereafter, in the ith super time slot, we schedule paths in group $G_{(i-1)\%\omega}$. Taking the data collection tree shown in Figure 4 as an example, $p_1, p_2, \ldots, p_7$ can be divided into three groups with $G_0 = \{p_1, p_4, p_7\}$, $G_1 = \{p_2, p_5\}$, and $G_2 = \{p_3, p_6\}$. Thereafter, G_0, G_1, and G_2 will be scheduled in a round-robin fashion. Within a group, the supernodes on all the paths can also be divided into at most ω node groups $g_k (0 \le k \le \omega - 1)$, with each node group containing mutually compatible nodes. Then, in the jth available super time slot for a particular node-group, we schedule the supernodes in $g_{(j-1)\%\omega}$. For group G_0 in the previous example, the supernodes on paths p_1, p_4, and p_7 can be divided into three node groups, and they can be scheduled in a round-robin manner in the available super time slots for G_0.

Step 2 (line 8). Schedule paths $p'_2, p'_3, \ldots, p'_{\lambda'}$ until all the data packets on these paths have been transmitted to the supernodes on p_h. This step can be done in a similar way as in Step 1.

Step 3 (lines 9–12). Schedule path p_v until all the data packets have been transmitted to the sink. After Step 1, for any supernode $v_{\lambda',j} (1 \le j \le \lambda')$, it has the data of j supernodes. Then, we abstract p_v to a *virtual tree* rooted at the sink, having λ' internal disjoint paths (except at the root) with lengths $1, 2, \ldots, \lambda'$, respectively, by splitting supernode $v_{\lambda',j}$ into j *virtual nodes*. Now, in the virtual tree, every virtual node contains exactly the same data with a supernode as the result of the splitting. For instance, p_v in Figure 4 is abstracted to a virtual tree shown in Figure 5. Afterwards, we schedule each path of the resulting virtual tree by a similar path-scheduling method used in Step 1.

Step 4 (line 13). Schedule path p_h until all the data packets have been transmitted to the sink. This step can be done in a similar way as in Step 3.

Consequently, after these four steps of CBPS, the data of a snapshot can be collected by the sink.

$\kappa_{i,j}$ can transmit $8 \log n$ data packets to the sensors in $\kappa_{i',j'}$ in terms of some order, for example, the sensor with small ID first.

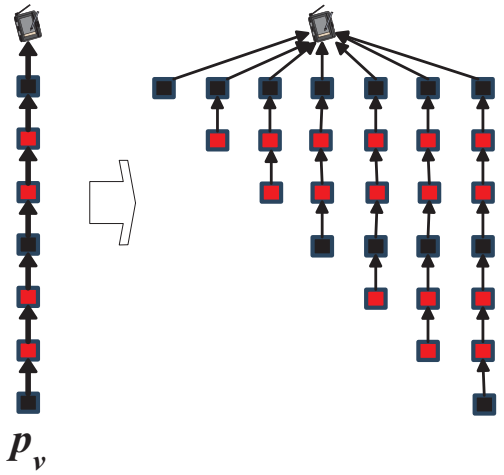


Fig. 5. A virtual tree.

4.2. Network Capacity Analysis of CBPS

In this section, we derive the achievable network capacity of CBPS. The upper bound of the SDC capacity is W, which has been explained [Chen et al. 2009b]. Consequently, we focus on the lower bound of CBPS.

In each of the four steps of CBPS, the basic scheduling blocks are paths. Hence, we show the upper bound of the number of time slots used to schedule a path in the following lemma.

LEMMA 4.1. *For a path p (p consists of supernodes) of length L, it takes $8\omega L \log n$ time slots to collect all the data packets on p by the sink.*

PROOF. From the description of CBPS, we know that the supernodes (or virtual nodes) on p can be divided into at most ω compatible nodegroups. In each available super time slot for p, we can schedule the supernodes in a compatible node group. Therefore, after ω super time slots, all the supernodes on p have been scheduled once. Hence, all the supernodes have transmitted their data to their corresponding parent supernodes except for the sink (or the last end supernode of p), and all the supernodes except for the leaf supernode have received the data from their corresponding children supernodes. Therefore, after every ω super time slots, the data transmission path decreases by one. Finally, the data on p can be collected by the sink within $\omega + (L-1)\cdot\omega$ super time slots, that is, $8\omega L \log n$ time slots. □

From Lemma 4.1, we have the following corollary which shows the number of time slots used to collect data on p_1 (p_1 is the path corresponding to the cells on the principal diagonal).

COROLLARY 4.2. *The data on p_1 can be collected to the cells on p_v (i.e., the sink) within $8\omega\lambda' \log n$ time slots.*

LEMMA 4.3. *Step 1 of CBPS can be finished within $8\omega^2\lambda' \log n$ time slots.*

PROOF. In Step 1, the paths in a compatible path group can be scheduled simultaneously, and the compatible path groups are scheduled in a round-robin fashion in terms of super time slots. Therefore, the number of time slots used in Step 1 depends on the compatible path group with the longest path. p_1 is the longest path, and the compatible

path group containing p_1 is scheduled every ω super time slots. Furthermore, by Corollary 4.2, the number of time slots used to collect the data on p_1 is at most $8\omega\lambda' \log n$. Hence, it follows that the number of time slots used to collect data on $p_1, p_2, \ldots, p_{\lambda'}$ is at most $8\omega^2\lambda' \log n$. □

From Lemma 4.3, we have the following corollary.

COROLLARY 4.4. *Step 1 and Step 2 of CBPS can be finished within* $16\omega^2\lambda' \log n$ *time slots.*

LEMMA 4.5. *Step 3 of CBPS can be finished within* $4\omega\lambda'(\lambda'+1)\log n$ *time slots.*

PROOF. According to CBPS, path p_v is abstracted into a virtual tree having λ' internally disjoint paths (except at the sink). Furthermore, the lengths of these λ' paths are $1, 2, \ldots, \lambda'$, respectively. By Lemma 4.1, the number of time slots used to collect data on these λ' paths is at most

$$
\begin{aligned}
& 8\omega \cdot 1 \cdot \log n + 8\omega \cdot 2 \cdot \log n + \cdots + 8\omega \cdot \lambda' \cdot \log n && (1)\\
&= 8\omega \cdot \log n \cdot (1 + 2 + \cdots + \lambda') && (2)\\
&= 8\omega \cdot \log n \cdot \frac{\lambda'(\lambda'+1)}{2} && (3)\\
&= 4\omega\lambda'(\lambda'+1)\log n. && (4)
\end{aligned}
$$

□

From Lemma 4.5, we have the following corollary.

COROLLARY 4.6. *Step 3 and Step 4 of CBPS can be finished within* $8\omega\lambda'(\lambda'+1)\log n$ *time slots.*

For the entire CBPS algorithm, the number of time slots is bounded by $O(\frac{1}{4\omega} \cdot n)$, which is proved in Theorem 4.7.

THEOREM 4.7. *The number of time slots used by CBPS to collect a snapshot data is bounded by* $O(4\omega n)$.

PROOF. Suppose T is the number of time slots used by CBPS to collect snapshot data by the sink. Then, by Corollaries 4.4 and 4.6, we have

$$
\begin{aligned}
T &\le 16\omega^2\lambda' \log n + 8\omega\lambda'(\lambda'+1)\log n && (5)\\
&\le 16\omega^2\lambda \log n + 8\omega\lambda^2 \log n && (6)\\
&= 16\omega^2 \cdot \frac{\sqrt{cn}}{\sqrt{2c\log n}} \cdot \log n + 8\omega \cdot \left(\frac{\sqrt{cn}}{\sqrt{2c\log n}}\right)^2 \cdot \log n && (7)\\
&= 16\omega^2 \cdot \sqrt{\frac{n\log n}{2}} + 4\omega n && (8)\\
&\le O(4\omega n). && (9)
\end{aligned}
$$

□

Now, we can obtain the lower bound of the achievable network capacity of CBPS, which is order-optimal, as shown in Theorem 4.8.

THEOREM 4.8. *The achievable network capacity of CBPS is* $\Omega(\frac{1}{4\omega} \cdot W)$, *which is order-optimal.*

PROOF. By Theorem 4.7 and the definition of network capacity, we have $C = \frac{n \cdot b}{\tau} \geq \frac{n \cdot b}{O(4\omega n) \cdot t} = \Omega(\frac{1}{4\omega} \cdot W)$. Since it has been proved that the upper bound of the data collection capacity is W, this implies that CBPS is order-optimal. □

When addressing the CDC problem, an intuitive idea is to pipeline the existing SDC algorithms. However, such an idea cannot improve the achievable network capacity in order, as explained in Ji et al. [2011], because data transmissions at the nodes far from the sink can really be accelerated by a pipeline. Nevertheless, the fact that a sink can receive at most one packet during each time slot makes the data accumulated at the nodes near the sink Ji et al. [2011].

5. NETWORK CAPACITY OF CONTINUOUS DATA COLLECTION

Since most of the existing work with pipeline technology cannot improve the CDC capacity significantly, as discussed in Section 4, in this section, we propose a novel *Segment-Based Pipeline Scheduling* (SBPS) algorithm based on the technology used in compressive data gathering (CDG) [Luo et al. 2009] for CDC in WSNs. Theoretical analysis shows that the proposed SBPS algorithm can achieve a surprising network capacity.

5.1. Segment-Based Pipeline Scheduling (SBPS)

CDG was first proposed in Luo et al. [2009] for snapshot data gathering in single-radio single-channel WSNs. The basic idea of CDG is to distribute the data collection load uniformly to all the nodes in the entire network. We take the data collection on a path consisting of L sensors $s_1, s_2, \ldots, s_L$ and one sink s_0, as shown in Figure 6 [Luo et al. 2009], as an example to explain CDG. In Figure 6, the packet produced at sensor $s_j (1 \leq j \leq L)$ is d_j. For the basic data collection shown in Figure 6(a), s_1 transmits one packet d_1 to s_2, s_2 transmits two packets d_1 and d_2 to s_3, and finally all the packets on the path are transmitted to s_0 by s_L. To balance the transmission load, Luo et al. [2009] proposed the CDG method, as shown in Figure 6(b). Instead of transmitting the original data directly, s_1 multiplies its data with a random coefficient $\phi_{i1} (1 \leq i \leq M)$, and sends the M results $\phi_{i1} d_1$ to s_2. Upon receiving $\phi_{i1} d_1 (1 \leq i \leq M)$ from s_1, s_2 multiplies its data d_2 with a random coefficient $\phi_{i2} (1 \leq i \leq M)$, adds it to $\phi_{i1} d_1$, and then sends $\phi_{i1} d_1 + \phi_{i2} d_2$ as one data packet to s_3. Finally, s_L does a similar multiplication and addition and sends the result $\sum_{j=1}^{L} \phi_{ij} d_j (1 \leq i \leq M)$ to s_0. After s_0 receives all M packets, s_0 can restore the original packets based on the compressive sampling theory [Luo et al. 2009]. The number of the transmitted packets is $O(n^2)$ in Figure 6(a) and is $O(nM)$ in Figure 6(b), and usually $M \ll n$ for large-scale WSNs. Therefore, CDG reduces the number of transmitted packets.

Thanks to the benefit brought by CDG, we can address the CDC problem with the pipeline technique. Since we partition the network into interference zones $o_{i,j} (1 \leq i, j \leq \lceil \sqrt{cn}/R \rceil)$ in Section 3.4, we here define a new term called *segment* based on interference zones. On the basis of interference zone $o_{i,i} (1 \leq i \leq \lceil \sqrt{cn}/R \rceil)$, the area consisting of interference zones $o_{j,i} (i \leq j \leq \lceil \sqrt{cn}/R \rceil)$ and $o_{i,j} (i \leq j \leq \lceil \sqrt{cn}/R \rceil)$ is called a *segment*, denoted by S_i. Taking the network shown in Figure 7 as an example, there are three segments S_1 (consisting of interference zones $\{o_{1,1}, o_{2,1}, o_{3,1}, o_{1,2}, o_{1,3}\}$), S_2 (consisting of interference zones $\{o_{2,2}, o_{3,2}, o_{2,3}\}$), and S_3 (consisting of interference zone $\{o_{3,3}\}$) in the network, as shown in Figure 8(a). Within a segment S_i, the area consisting of cells on and below the principal diagonal is denoted by S_{ir}, and the area consisting of the remaining cells is denoted by S_{iu}, that is, $S_i = (S_{ir}, S_{iu})$. For instance, in the network shown in Figure 7, $S_{1r} = \{\kappa_{i,1} (1 \leq i \leq \lambda), \kappa_{i,2} (2 \leq i \leq \lambda), \kappa_{i,3} (3 \leq i \leq \lambda)\}$, and $S_{1u} = \{\kappa_{1,j} (1 < j \leq \lambda), \kappa_{2,j} (2 < j \leq \lambda), \kappa_{3,j} (3 < j \leq \lambda)\}$.

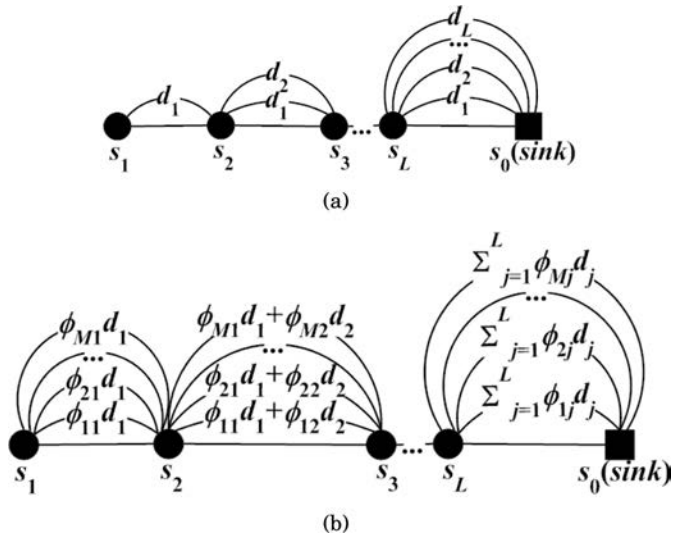


Fig. 6. Comparison of (a) basic data collection and (b) CDG [Luo et al. 2009].

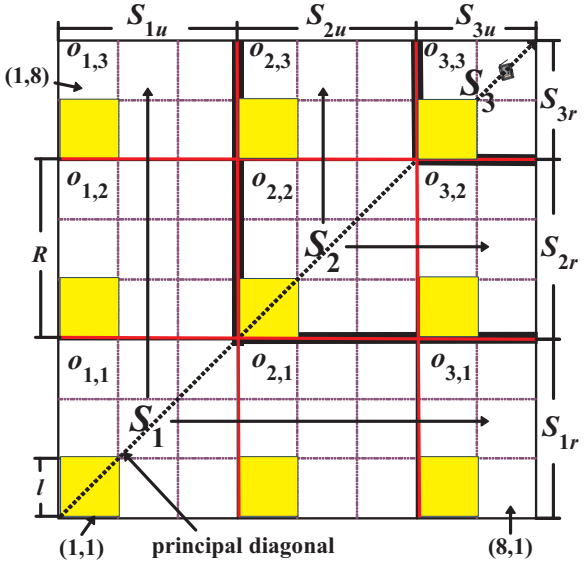


Fig. 7. Segments.

For CDC, we use a similar routing structure as in the CBPS algorithm (note it does not imply the same scheduling), that is, we abstract each cell as a supernode and then construct a data collection tree following the same rules as in CBPS (we use cells and supernodes interchangeably in the subsequent discussion). Unlike in CBPS, a supernode here can compress its currently-held data packets of a snapshot into M data packets for transmission.

Based on the defined segments and the constructed data collection tree, we propose a Segment-Based Pipeline Scheduling (SBPS) algorithm for CDC. Basically, SBPS schedules a CDC task by forming a data transmission pipeline on the segments, with each

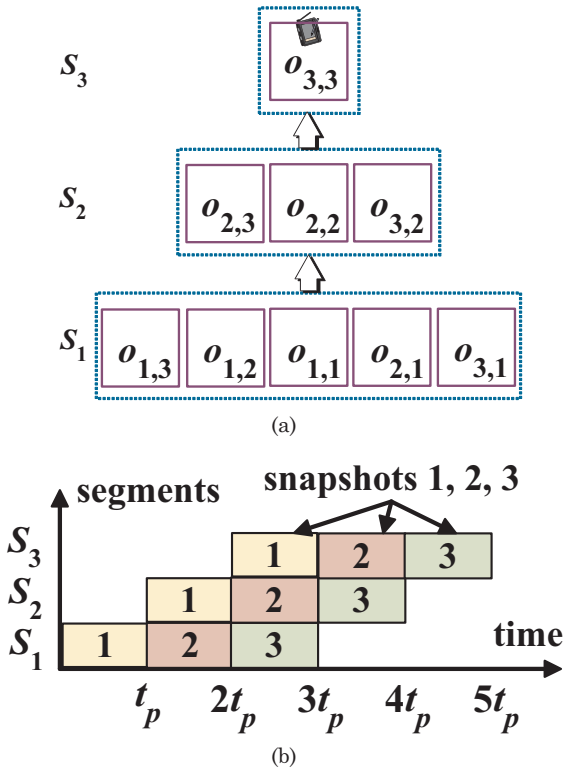


Fig. 8. Data collection pipeline of three segments.

segment as a pipeline unit. For example, Figure 8(a) is a pipeline system consisting of three segments (i.e., pipeline units), and Figure 8(b) shows a data transmission pipeline formed on these three segments. To schedule the segments to form a data transmission pipeline, we take the similar idea as in CBPS to partition cells into compatible cell groups, with each group consisting of all the compatible cells. Consequently, if all the cells transmit data in the CDG manner, a macroscopic data transmission pipeline can be formed when we schedule these compatible cell groups sequentially and repeatedly. We present the detailed idea of SBPS in a hierarchy-level fashion as follows.

First, scheduling at the segment level. The basic idea is to form a data transmission pipeline on the segments (pipeline units). At this level, each segment as a whole is considered. Since there is no intersection between any two segments, we can pipeline the data transmission on the segments (it can also be guaranteed that there is no wireless interference among segments in the next step), that is, for each segment $S_i = (S_{ir}, S_{iu})$, S_i starts the data transmission of the $(k+1)$-th snapshot immediately after it transmits all the data of the k-th snapshot to segment S_{i+1}. Let $t_p = \max\{t(S_i)|1 \le i \le \lceil \sqrt{cn}/R \rceil, t(S_i)$ *be the number of time slots used by segment* S_i *to transmit all the data packets of a snapshot*$\}$. Then, a segment data transmission pipeline on all the segments is formed with each segment working with t_p time slots for every snapshot (here, a snapshot is an individual task in a traditional pipeline operation). For instance, the network shown in Figure 7 can be partitioned into three segments S_1, S_2, and S_3 as shown in Figure 8(a). Therefore, a data transmission pipeline can be formed on these segments by a segment transmitting the data of a previous snapshot and

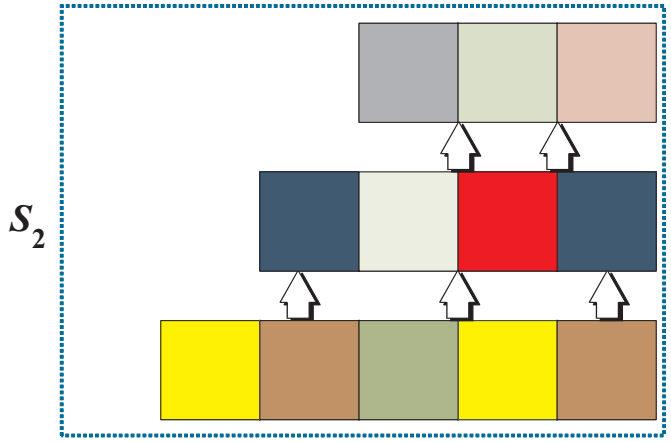


Fig. 9. Scheduling within a segment. (Cells with the same color are compatible cells, which can be scheduled concurrently).

receiving the data of a subsequent snapshot simultaneously. Figure 8(b) shows the data transmission pipeline formed by the pipeline system shown in Figure 8(a) for collecting three snapshots. From Figure 8(b), we can see that from time $2t_p$ to time $3t_p$, segment S_2 transmits the data of snapshot 2 to segment S_3 and meanwhile receives the data of snapshot 3 from S_1. Therefore, the data collection speed can be accelerated because of the time overlap of the transmission of continuous snapshots.

Second, scheduling at the row / column level. The basic idea is to partition cells into compatible cell groups, with each group consisting of compatible cells. Then, these groups will be scheduled sequentially and repeatedly. For the kth snapshot, within each segment $S_i = (S_{ir}, S_{iu})$, we first schedule S_{ir} to transmit the data in the cells of S_{ir} to $S_{(i+1)r}$ row by row. For instance, for segment S_{2r} in Figure 7, the data transmission flow is shown in Figure 9. Thereafter, we schedule S_{iu} in a similar way to transmit the data in the cells of S_{iu} to $S_{(i+1)u}$ column by column. When we schedule S_{ir}, the first row of cells of S_{ir}, that is, the cells $\kappa_{j,i}(i \le j \le \lambda)$, are scheduled first, followed by the second row of cells, that is, the cells $\kappa_{j,i+1}(i+1 \le j \le \lambda)$, and so on, until the last row of cells of S_{ir}, that is, the cells $\kappa_{j,i+\omega-1}(i+\omega-1 \le j \le \lambda)$, are scheduled. When we schedule each row, we can divide the cells on that row into ω compatible cell groups $g_1^i, g_2^i, \ldots, g_\omega^i$ with each group containing mutually compatible cells. Afterwards, $g_1^i, g_2^i, \ldots, g_\omega^i$ are scheduled in sequence. For example, within the segment shown in Figure 9, we can partition the cells into nine compatible cell groups. Then, we will schedule these nine groups one by one from the bottom level to the upper level. Note that we follow the same approach when we schedule all the segments in the segment data transmission pipeline. Therefore, all the cells in $g_j^i(1 \le j \le \omega, 1 \le i \le \lceil \sqrt{cn}/R \rceil)$ are also mutually compatible cells, according to the discussion in Section 3.4. This implies that all the segments can be scheduled without wireless interference. Afterwards, we can schedule cells in S_{iu} column by column in a similar way. Finally, S_i transmits all the data packets of the kth snapshot to its subsequent segment S_{i+1}.

Third, scheduling at the cell level. The basic idea is that each cell transmits data according to the CDG manner. Therefore, each time to schedule a cell, we will assign that cell $8M \cdot \log n$ time slots to transmit data for a snapshot. Every sensor in cell $\kappa_{i,j}$, generates one data packet of the kth snapshot. Furthermore, the sensors in cells $\kappa_{i,1}(1 \le i \le \lambda)$ and $\kappa_{1,j}(1 \le j \le \lambda)$, will not receive any data packets of the kth snapshot

according to the previous segment-level and row/column-level scheduling strategies (actually, this is true for any snapshot). Thus, the sensors in $\kappa_{i,1}(1 \leq i \leq \lambda)$ and $\kappa_{1,j}(1 \leq j \leq \lambda)$ transmit the packets of the kth snapshot in the CDG way in their available time slots, that is, for each sensor, it multiplies its data with M random coefficients, respectively, and sends the newly obtained M products to its parent node. The sensors in $\kappa_{i,j}(1 < i, j \leq \lambda)$, will receive some data packets of the kth snapshot. (It is possible that some sensors do not have any children. In this case, they do the same operation as the sensors in $\kappa_{i,1}(1 \leq i \leq \lambda)$ and $\kappa_{1,j}(1 \leq j \leq \lambda)$.) After they receive all the packets of the kth snapshot from their children sensors, they combine their data and the received data in the same way as in CDG and transmit the obtained M data packets to their parent sensors, respectively. For the sink, it restores the data of a snapshot in the CDG way after it receives all the packets of that snapshot. Here it is straightforward that it takes at most $8M \cdot \log n$ time slots for a cell to transmit the data packets of a snapshot to the subsequent cell, since every cell contains at most $8 \log n$ sensors by Lemma 3.2.

5.2. Network Capacity Analysis of SBPS

We analyze the achievable network capacity of SBPS. Since t_p in the SBPS algorithm is essential for our analysis, we give the upper bound of t_p in the following lemma.

LEMMA 5.1. *For t_p in SBPS, $t_p \leq 16\omega^2 M \log n$.*

PROOF. It has been pointed out in Section 5.1 that it takes a cell at most $8M \cdot \log n$ time slots to transmit the data of a particular snapshot. Furthermore, when we schedule each row of a segment $S_i = (S_{ir}, S_{iu})$, we divide the cells of that row into ω compatible cell groups and schedule these groups in sequence. Therefore, the data packets of a particular snapshot contained in the cells of a row can be transmitted to the subsequent row within $8\omega M \log n$ time slots. Each segment has at most ω rows. Therefore, the number of time slots used to schedule cells in S_{ir} is at most $8\omega^2 M \log n$. For the same reason, the number of time slots used to schedule the cells in S_{iu} for a particular snapshot is also at most $8\omega^2 M \log n$. In a sum, $t_p \leq 16\omega^2 M \log n$. □

Based on Lemma 5.1, we obtain the upper bound of the number of time slots used by SBPS to collect N continuous snapshots as follows.

THEOREM 5.2. *The number of time slots used by the SBPS algorithm to collect N continuous snapshots is at most $8\omega M\sqrt{2n \log n} + 16\omega^2 MN \log n$.*

PROOF. Suppose the number of time slots used by SBPS to collect N continuous snapshots is T. By Lemma 5.1, it costs a segment $16\omega^2 M \log n$ time slots to transmit the data of a snapshot to the subsequent segment. Afterwards, that segment starts to transmit data for the following snapshot immediately, according to SBPS. Therefore, by the formed segment data transmission pipeline, the sink can collect the data of a snapshot every $16\omega^2 M \log n$ time slots after it receives the data of the first snapshot. Thus, to receive the data of N continuous snapshots, we have

$$T \leq 16\omega^2 M \log n \cdot \left\lceil \frac{\lambda}{\omega} \right\rceil + (N-1) \cdot 16\omega^2 M \log n \tag{10}$$

$$\leq 16\omega M \lambda \log n + 16\omega^2 MN \log n \tag{11}$$

$$= 16\omega M \sqrt{\frac{cn}{2c \log n}} \log n + 16\omega^2 MN \log n \tag{12}$$

$$= 8\omega M\sqrt{2n \log n} + 16\omega^2 MN \log n. \tag{13}$$

□

Now, we are prepared to derive the achievable network capacity of SBPS for CDC as shown in Theorem 5.3.

THEOREM 5.3. *The achievable network capacity of the SBPS algorithm is* $\Omega(\sqrt{\frac{n}{\log n}}W)$ *when* $N \leq \sqrt{\frac{n}{\log n}}$*; or* $\Omega(\frac{n}{\log n}W)$ *when* $N > \sqrt{\frac{n}{\log n}}$.

PROOF. By Theorem 5.2, it takes the SBPS algorithm at most $8\omega M\sqrt{2n\log n} + 16\omega^2 MN\log n$ time slots to collect N continuous snapshots by the sink. Then, we discuss the achievable network capacity of SBPS case by case.

Case 1. $N \leq \sqrt{\frac{n}{\log n}}$. In this case,

$$T \leq 8\omega M\sqrt{2n\log n} + 16\omega^2 MN\log n \tag{14}$$

$$\leq O(8\omega M\sqrt{2n\log n}). \tag{15}$$

Thus, we have

$$C = \frac{nN\cdot b}{\tau} \tag{16}$$

$$\geq \frac{nN\cdot b}{O(8\omega M\sqrt{2n\log n})\cdot t} \tag{17}$$

$$= \Omega\left(\sqrt{\frac{n}{\log n}}W\right), \tag{18}$$

since ω is a constant value and $M \ll n$.

Case 2. $N > \sqrt{\frac{n}{\log n}}$. In this case,

$$T \leq 8\omega M\sqrt{2n\log n} + 16\omega^2 MN\log n \tag{19}$$

$$\leq O(16\omega^2 MN\log n). \tag{20}$$

Thus, we have

$$C = \frac{nN\cdot b}{\tau} \tag{21}$$

$$\geq \frac{nN\cdot b}{O(16\omega^2 MN\log n)\cdot t} \tag{22}$$

$$= \Omega\left(\frac{n}{\log n}W\right). \tag{23}$$

□

From Theorem 5.3, the proposed scheduling algorithm SBPS can achieve a high network capacity by combining the pipeline and CDG techniques. Since the current best result is $\Omega(W)$ Chen et al. [2009b], our result is at least $\sqrt{\frac{n}{\log n}}$ or $\frac{n}{\log n}$ times better than the current best result, which is a very significant improvement. Actually, there are two main reasons for the surprising results. One is the pipeline scheduling. By Theorem 5.2, the time overlap of the collection of multiple continuous snapshots on the segment data transmission pipeline takes a large number of time slots, which accelerates the data collection process directly and significantly. Another reason is the CDG method. As indicated in Luo et al. [2009], CDG can reduce the total number of packet transmissions.

6. SIMULATIONS AND ANALYSIS

In this section, we validate the performances of the proposed algorithms via simulations. The network settings are determined based on the network model defined in

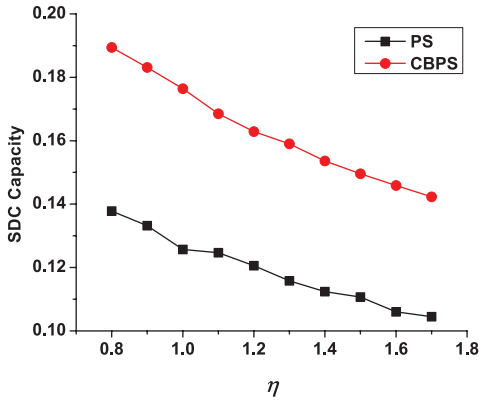


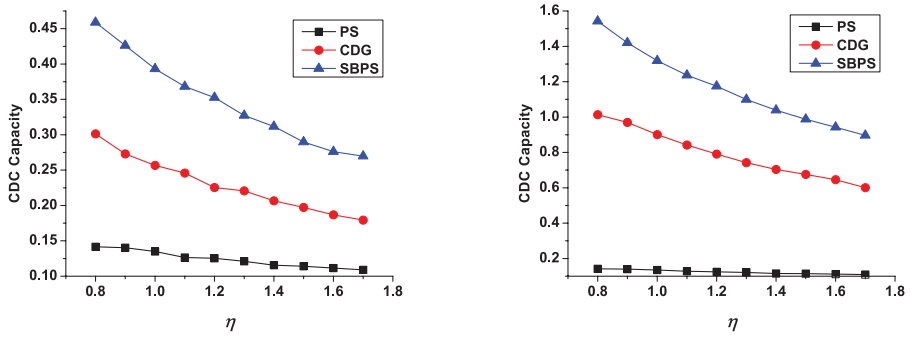
Fig. 10. SDC capacity.

Section 3.1. To be detailed, for all the simulations, we consider a WSN with one sink, where all the sensor nodes are randomly distributed in a square area of size $X \times Y$. The network time is slotted, and the length of each time slot is assumed to be one second. Every node produces one data packet in a snapshot, and the size of a packet is normalized to one. All the nodes work with a fixed power P over a common wireless channel whose bandwidth is also normalized to one. This implies that a sensor can transmit one data packet during a time slot if there is no interference. Furthermore, we define the average number of nodes distributed within a unit area as the *node density* of a WSN, denoted by ρ. In all the following simulations, we set the path-loss exponent $\alpha = 3.0$, the node density $\rho = 3.0$ (which implies there are $3XY$ sensor nodes in a network of size $X \times Y$), and $\frac{P}{N_0} = 10.0$. For CDC, we set the parameter M in CDG as $M = 50$ and the number of snapshots in a CDC task as $N = 1{,}000$. After deploying the network, we adopt the proposed network partition method to partition the network into cells and determine some important network parameters, for example, ω, R, etc. Subsequently, we run the proposed SDC algorithm and CDC algorithm, and compared them with existing works.

Particularly, we compared CBPS with PS [Chen et al. 2010] and SBPS with CDG [Luo et al. 2009] and PS. PS is the latest data collection algorithm based on a breadth-first search (BFS) tree. CDG is a recent work for data collection and the first work applying the compressive sampling theory. The idea of CDG has been discussed in Section 5. When comparing PS and CDG with SBPS, we also apply the pipeline technique on them for fairness. Each group of the following simulations is repeated 100 times and the results are the average values.

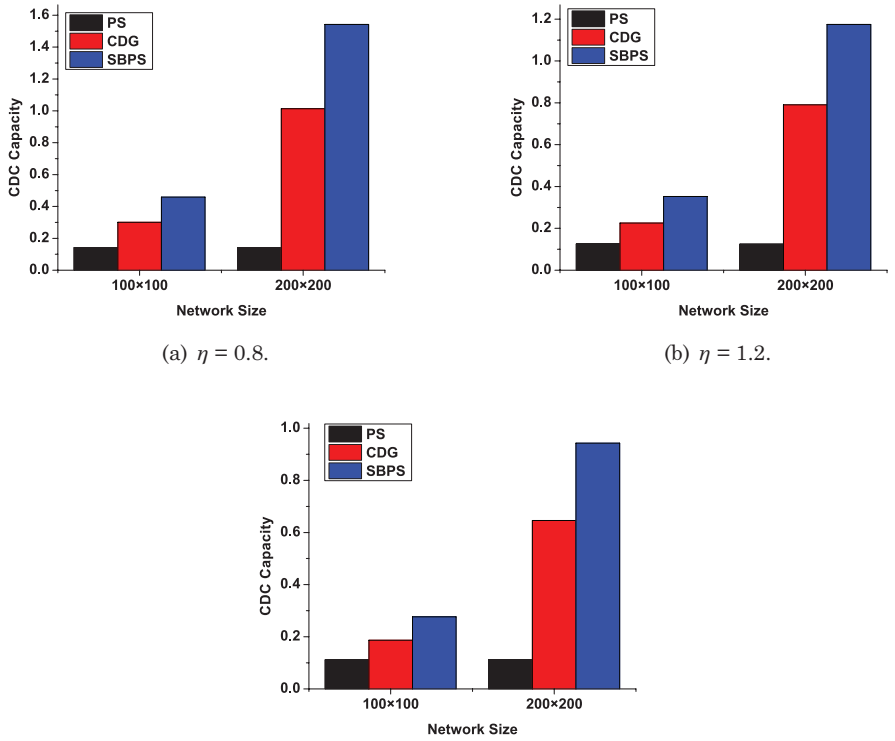
6.1. Performance of CBPS

We implement PS and CBPS for SDC in a WSN deployed in an area of 100×100. The achievable network capacities are shown in Figure 10. From Figure 10, we can see that with the increase of η, the capacities of both PS and CBPS decrease. This is because a large η implies that high-quality signal (or stronger signal strength) is required to successfully receive a data packet. This can also be seen from Theorem 3.3, that a large η leads to a large R, which further implies that a fewer number of compatible cells (nodes) can conduct transmissions concurrently. As a result, the achievable capacities of PS and CBPS decrease as the data transmission concurrency decreases. Moreover, since CBPS schedules multiple compatible cells on multiple data transmission paths simultaneously, unlike PS which only schedules the nodes on a single path, CBPS



(a) Network size is 100 × 100. (b) Network size is 200 × 200.

Fig. 11. CDC capacity with respect to η.



(a) $\eta = 0.8$. (b) $\eta = 1.2$.

(c) $\eta = 1.6$.

Fig. 12. CDC capacity with respect to network size.

always achieves a higher network capacity than PS. On average, CBPS achieves 36.9% more capacity than PS.

6.2. Performance of SBPS

To examine the performance of SBPS, we compare its achievable capacity for CDC with PS with pipeline and CDG with pipeline in WSNs deployed in areas of 100 × 100 and 200 × 200, respectively. The results are shown in Figure 11 and Figure 12. From

Figure 11, we can see that the achievable capacities of PS, CDG, and SBPS decrease with the increase of η, because (i) a high η (which will induce a a large R according to Theorem 3.3) implies that fewer compatible cells (nodes) can concurrently transmit data; and (ii) a large R implies larger segments and large t_p (defined in Section 5). Therefore, the pipeline capacity is decreased, which further leads to the decrease of the capacities of CDG and SBPS.

From Figure 12, we can also see that network size (i.e., the number of sensors in a WSN) impacts the capacities of pipeline-based CDG and SBPS. This is because data transmission pipeline is easier to form and more effective in WSNs with larger sizes, as shown in the analysis of SBPS. By contrast, network size has little impact on the performance of PS due to the data accumulation problem, as explained in Section 4. On the other hand, since both CDG and SBPS have addressed the data accumulation problem, they can form effective data transmission pipelines, which ultimately makes them achieve higher capacities than PS.

Furthermore, from Figures 11 and 12, SBPS also achieves a higher capacity than CDG. This is because (i) the data collection tree used by SBPS is balanced and has a more preferable structure by a pipeline; and (ii) SBPS has a more sound scheduling scheme, which can achieve complete concurrency by scheduling multiple compatible cells simultaneously. Particularly, in a WSN of size 200×200, on average, SBPS achieves a capacity which is 9.39 times that of PS, and 47.8% more than that of CDG.

7. CONCLUSION AND FUTURE WORK

Most existing works focus on network capacity of unicast, multicast or/and broadcast, which are different communication modes from data collection, especially CDC. In this article, we first study the SDC problem under PhIM and propose a Cell-Based Path Scheduling (CBPS) algorithm based on network partitions. Theoretical analysis of CBPS shows that its achievable network capacity is $\Omega(W)$, which is order-optimal. For CDC, we investigate the behind-the-scene reasons for this limitation. Thereafter, we propose a novel Segment-Based Pipeline Scheduling (SBPS) algorithm. SBPS significantly speeds up the CDC process and achieves a surprising network capacity, which is at least $\sqrt{\frac{n}{\log n}}$ or $\frac{n}{\log n}$ times better than the current best result. Furthermore, the simulation results also validate that the proposed algorithms significantly improve network capacity compared with existing works.

The future work of this article can be conducted according to the following directions. First, instead of assuming all the nodes are randomly deployed, we will study the achievable capacity of WSNs where the nodes are arbitrarily deployed. Second, since most of the existing works that study the network capacity issue are for centralized WSNs, we will investigate the achievable data collection capacity in a distributed manner.

APPENDIX

A. PROOF OF LEMMA 3.1

PROOF (LEMMA 3.1). Since the sensors are i.i.d., the probability of any particular cell $\kappa_{i,j}$ to being empty is $(1-\frac{l^2}{A})^n$, that is, $\Pr[e_{ij}] = (1-\frac{l^2}{A})^n$ [Kulkarni and Viswanath 2004]. Considering the fact that there are λ^2 cells in the network and by Boole's inequality, we have

$$\Pr\left[\bigcup_{1\le i,j\le\lambda} e_{ij}\right] \le \sum_{1\le i,j\le\lambda} \Pr[e_{ij}] \tag{24}$$

$$= \lambda^2 \left(1 - \frac{l^2}{A}\right)^n \tag{25}$$

$$= \frac{n}{2\log n}\left(1 - \frac{2\log n}{n}\right)^n \tag{26}$$

$$\leq \frac{n}{2\log n} \cdot \exp(-2\log n) \tag{27}$$

$$\leq \frac{n}{2\log n} \cdot \exp(-2\ln n) \tag{28}$$

$$= \frac{n}{2\log n} \cdot \frac{1}{n^2} \tag{29}$$

$$= \frac{1}{2n\log n}. \tag{30}$$

□

B. PROOF OF LEMMA 3.2

PROOF (LEMMA 3.2). Since all the sensors are i.i.d., the number of sensors in a cell satisfies the binomial distributions with parameters $(n, \frac{l^2}{A})$ [Kulkarni and Viswanath 2004]. Applying the Chernoff bound and for any $\gamma > 0$, we have

$$\Pr[Z_{ij} > 8\log n] \leq \frac{\mathrm{E}[\exp(\gamma Z_{ij})]}{\exp(8\gamma\log n)} \tag{31}$$

$$= \frac{\left[1 + (\exp(\gamma) - 1)\frac{l^2}{A}\right]^n}{\exp(8\gamma\log n)} \tag{32}$$

$$= \frac{\left[1 + \frac{2\log n}{n}(\exp(\gamma) - 1)\right]^n}{\exp(8\gamma\log n)} \tag{33}$$

$$\leq \frac{exp(2\log n(\exp(\gamma) - 1))}{\exp(8\gamma\log n)} \tag{34}$$

$$= \exp((2\exp(\gamma) - 8\gamma - 2) \cdot \log n). \tag{35}$$

Let $\gamma = \frac{1}{2}$, then

$$\Pr[Z_{ij} > 8\log n] \leq \exp((2\sqrt{e} - 6) \cdot \log n) \tag{36}$$

$$\leq \exp(-2\log n) \tag{37}$$

$$\leq \exp(-2\ln n) \tag{38}$$

$$= \frac{1}{n^2}. \tag{39}$$

□

C. PROOF OF THEOREM 3.3

PROOF (THEOREM 3.3). Let $\mathcal{C}$ be a *compatible cell set* that contains any cell $\kappa_{i,j}$ and all of its compatible cells, that is, $\mathcal{C} = \{\kappa_{i,j}\} \cup \{\kappa_{i',j'} | \kappa_{i',j'} \neq \kappa_{i,j}$, and $\kappa_{i',j'}$ is a compatible cell of $\kappa_{i,j}\}$. To make every cell in $\mathcal{C}$ able to initiate a data transmission concurrently

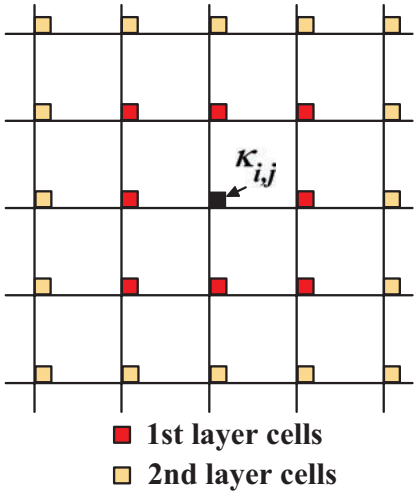


Fig. 13. Computation of R.

without interference[6], it is sufficient to have

$$\frac{P \cdot \|\kappa_{i,j} - \kappa'_{i,j}\|^{-\alpha}}{N_0 + \sum_{\kappa_{i',j'} \in \mathcal{C}, \kappa_{i',j'} \neq \kappa_{i,j}} P \cdot \|\kappa_{i',j'} - \kappa'_{i,j}\|^{-\alpha}} \geq \eta, \tag{40}$$

where $\kappa_{i,j}$ is any transmitting cell, $\kappa'_{i,j}$ is the receiving cell of $\kappa_{i,j}$, and $\kappa_{i',j'}$ is a compatible cell of $\kappa_{i,j}$ ($\kappa_{i',j'}$ is also transmitting some data packets simultaneously with $\kappa_{i,j}$). Since P, N_0, and η are constant values, we derive the bounds of $\|\kappa_{i,j} - \kappa'_{i,j}\|^{-\alpha}$ and $\sum_{\kappa_{i',j'} \in \mathcal{C}, \kappa_{i',j'} \neq \kappa_{i,j}} \|\kappa_{i',j'} - \kappa'_{i,j}\|^{-\alpha}$ in the following.

First, we have $\|\kappa_{i,j} - \kappa'_{i,j}\|^{-\alpha} \geq r^{-\alpha}$, since r is the transmission radius of a sensor (defined in Section 3.1) and every communication pair must be within the communication range of each other. Subsequently, all the compatible cells of $\kappa_{i,j}$ in $\mathcal{C}$ can be layered with respect to $\kappa_{i,j}$, as shown in Figure 13, with the ψth ($\psi \geq 1$) layer having at most 8ψ cells.[7] Furthermore, the distance between the receiving cell of $\kappa_{i,j}$, that is, $\kappa'_{i,j}$, and any compatible cell at the ψth layer is no less than $\psi \cdot R - 2l$. Thus, we have

$$\sum_{\kappa_{i',j'} \in \mathcal{C}, \kappa_{i',j'} \neq \kappa_{i,j}} \|\kappa_{i',j'} - \kappa'_{i,j}\|^{-\alpha} \tag{41}$$

$$\leq \sum_{\psi \geq 1} 8\psi \cdot (\psi \cdot R - 2l)^{-\alpha} \tag{42}$$

$$= 8R^{-\alpha} \cdot \sum_{\psi \geq 1} \psi \left(\psi - \frac{2l}{R}\right)^{-\alpha} \tag{43}$$

[6]Here, we actually mean "to make every cell in $\mathcal{C}$ having a sensor able to initiate a data transmission concurrently without interference." Without confusion, we use cell and sensor interchangeably.
[7]This can be easily proven by mathematical induction.

$$\leq 8R^{-\alpha} \cdot \left[\left(1 - \frac{2}{3}\right)^{-\alpha} + \sum_{\psi \geq 2} \psi \left(\psi - \frac{2}{3}\right)^{-\alpha} \right] \tag{44}$$

$$\leq 8R^{-\alpha} \cdot \left[3^{\alpha} + \sum_{\psi \geq 2} \psi(\psi - 1)^{-\alpha} \right] \tag{45}$$

$$= 8R^{-\alpha} \cdot \left[3^{\alpha} + \sum_{\varpi \geq 1} \varpi^{-\alpha}(\varpi + 1) \right] \tag{46}$$

$$\leq 8R^{-\alpha} \cdot \left[3^{\alpha} + \sum_{\varpi \geq 1} \varpi^{-2} + \sum_{\varpi \geq 1} \varpi^{-3} \right] \tag{47}$$

$$= 8R^{-\alpha} \cdot (3^{\alpha} + \zeta(2) + \zeta(3)). \tag{48}$$

The transition from Eq. (45) to Eq. (46) is due to that we use ϖ to substitute $\psi - 1$, that is, $\varpi = \psi - 1$. The transition from Eq. (46) to Eq. (47) is because $\alpha \geq 3$. In Eq. (48), $\zeta(\cdot)$ is the *Riemann zeta function*, and with $\zeta(2) = \frac{\pi^2}{6} \approx 1.645$ and $\zeta(3) \approx 1.202$, respectively. Let $c_1 = 8(3^{\alpha} + 2.847)$. We have

$$\sum_{\kappa_{i',j'} \in \mathcal{C}, \kappa_{i',j'} \neq \kappa_{i,j}} \|\kappa_{i',j'} - \kappa'_{i,j}\|^{-\alpha} \leq c_1 \cdot R^{-\alpha}. \tag{49}$$

It follows that,

$$\frac{P \cdot \|\kappa_{i,j} - \kappa'_{i,j}\|^{-\alpha}}{N_0 + \sum\limits_{\kappa_{i',j'} \in \mathcal{C}, \kappa_{i',j'} \neq \kappa_{i,j}} P \cdot \|\kappa_{i',j'} - \kappa'_{i,j}\|^{-\alpha}} \geq \frac{P \cdot r^{-\alpha}}{N_0 + c_1 P \cdot R^{-\alpha}}. \tag{50}$$

Now, to make the inequality in Eq. (40) valid, it is sufficient to have

$$\frac{P \cdot r^{-\alpha}}{N_0 + c_1 P \cdot R^{-\alpha}} \geq \eta \tag{51}$$

$$\Leftrightarrow R \geq \left(\frac{r^{-\alpha}}{c_1 \eta} - \frac{N_0}{c_1 P} \right)^{-1/\alpha}. \tag{52}$$

Since N_0, α, c_1, η, and P are constant values, we have

$$R \geq \left(\frac{r^{-\alpha}}{c_1 \eta} - \frac{N_0}{c_1 P} \right)^{-1/\alpha} \sim \Theta(r).$$

Furthermore, considering that N_0 is negligible compared with the interference brought on by concurrent transmitters [Chau et al. 2009], we can ignore N_0, that is, let $N_0 = 0$. Thus, we have

$$R \geq \sqrt[\alpha]{c_1 \eta} \cdot r = 2\sqrt{2} \cdot \sqrt[\alpha]{c_1 \eta} \cdot l,$$

since $r = 2\sqrt{2}l$. Based on the definition of interference zones, a small R implies more compatible cells, which further implies that more sensors can conduct transmissions concurrently; we set $R = 2\sqrt{2} \cdot \sqrt[\alpha]{c_1 \eta} \cdot l$, which is sufficient to make every cell in $\mathcal{C}$ able to initiate a data transmission concurrently without interference. Now, let $\omega = 2\sqrt{2} \cdot \sqrt[\alpha]{c_1 \eta}$, that is, $R = \omega \cdot l$. Evidently, ω is a constant value. It follows that Theorem 3.3 holds. □

REFERENCES

ANDREWS, M. AND DINITZ, M. 2009. Maximizing capacity in arbitrary wireless networks in the SINR model: Complexity and game theory. In *Proceedings of the 28th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'09)*. 1332–1340.

BHANDARI, V. AND VAIDYA, N. H. 2007. Connectivity and capacity of multi-channel wireless networks with channel switching constraints In *Proceedings of the 26th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'07)*. 785–793.

CHAFEKAR, D., LEVIN, D., KUMAR, V. S. A., MARATHE, M. V., PARTHASARATHY, S., AND SRINIVASAN, A. 2008. Capacity of asynchronous random-access scheduling in wireless networks. In *Proceedings of the 27th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'08)*. 13–18.

CAI, Z., JI, S., HE, J. (S.), AND BOURGEOIS, A. G. 2012. Optimal distributed data collection for asynchronous cognitive radio networks. In *Proceedings of the 32nd International Conference on Distributed Computing Systems (ICDCS'12)*. 245–254.

CHAU, C.-K., CHEN, M., AND LIEW, S. C. 2009. Capacity of large-scale CSMA wireless networks. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom'09)*. 97–108.

CHEN, S., WANG, Y., LI, X.-Y., AND SHI, X. 2009. Order-optimal data collection in wireless sensor networks: Delay and capacity. In *Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'09)*. 1–9.

CHEN, S., WANG, Y., LI, X.-Y., AND SHI, X. 2009. Data collection capacity of random-deployed wireless sensor networks. In *Proceedings of the Global Communications Conference (GLOBECOM'09)*. 1–6.

CHEN, S., TANG, S., HUANG, M., AND WANG, Y. 2010. Capacity of data collection in arbitrary wireless sensor networks. In *Proceedings of the 29th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'10)*. 356–360.

CHIPARA, O., LU, C., AND STANKOVIC, J. 2006. Dynamic conflict-free query scheduling for wireless sensor networks. In *Proceedings of the 14th IEEE International Conference on Network Protocols (ICNP'06)*. 321–331.

DAI, H.-N., NG, K.-W., WONG, R. C.-W., AND WU, M.-Y. 2008. On the capacity of multi-channel wireless networks using directional antennas. In *Proceedings of the 27th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'08)*. 628–636.

DUARTE-MELO, E. J. AND LIU, M. 2003. Data-gathering wireless sensor networks: Organization and capacity. *Comput. Netw. 43*, 4, 519–537.

FRANCESCHETTI, M., DOUSSE, O., TSE, D. N. C., AND THIRAN, P. 2007. Closing the gap in the capacity of wireless networks via percolation theory. *IEEE Trans. Inf. Theory 53*, 3, 1009–1018.

GAMAL, H. E. 2003. On the scaling laws of dense wireless sensor networks. *IEEE Trans. Inf. Theory 51*, 3, 1229–1234.

GARETTO, M., GIACCONE, P., AND LEONARDI, E. 2007. On the capacity of ad hoc wireless networks under general node mobility. In *Proceedings of the 26th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'07)*. 357–365.

GOUSSEVSKAIA, O., WATTENHOFER, R., HALLDORSSON, M. M., AND WELZL, E. 2009. Capacity of arbitrary wireless networks. In *Proceedings of the 28th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'09)*. 1872–1880.

GUPTA, P. AND JUMAR, P. R. 2000. The capacity of wireless networks. *IEEE Trans. Inf. Theory 46*, 2, 388–404.

HUANG, W., WANG, X., AND ZHANG, Q. 2010. Capacity scaling in mobile wireless ad hoc network with infrastructure support. In *Proceedings of the 30th International Conference on Distributed Computing Systems (ICDCS'10)*. 848–857.

JI, S., LI, Y., AND JIA, X., 2011. Capacity of dual-radio multi-channel wireless sensor networks for continuous data collection. In *Proceedings of the 30th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'11)*. 1062–1070.

JI, S., ULUAGAC, A. S., BEYAH, R., AND CAI, Z. 2012. Practical unicast and convergecast scheduling for cognitive radio networks. *J. Combin. Optim. 26*, 1, Springer, 161–177.

JI, S., CAI, Z., LI, Y., AND JIA, X. 2012. Continuous data collection capacity of dual-radio multi-channel wireless sensor networks. *IEEE Trans. Parall. Distrib. Syst. 23*, 10, 1844–1855.

JI, S., BEYAH, R., AND CAI, Z. 2012. Snapshot/continuous data collection capacity for large-scale probabilistic wireless sensor networks. In *Proceedings of the 31st IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'12)*. 25–30.

JI, S. AND CAI, Z. 2012. Distributed data collection and its capacity in asynchronous wireless sensor networks. In *Proceedings of the 31st IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'12)*. 2113–2121.

KESHAVARZ-HADDAD, A., RIBEIRO, V., AND RIEDI, R. 2006. Broadcast capacity in multihop wireless networks. In *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MobiCom'06)*. 239–250.

KULKARNI, S. R. AND VISWANATH, P. 2004. A deterministic approach to throughput scaling in wireless networks. *IEEE Trans. Inf. Theory 50*, 6, 1041–1049.

KUMAR, V., S. A., MARATHE, M. V., PARTHASARATHY, S., AND SRINIVASAN, A. 2005. Algorithmic aspects of capacity in wireless networks. In *Proceedings of the International Conference on Measurements and Modeling of Computer Systems (SIGMETRICS'05)*. 133–144.

KYASANUR, P. AND VAIDYA, N. H. 2005. Capacity of multi-channel wireless networks: Impact of number of channels and interfaces. In *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom'05)*. 45–57.

LI, P. AND FANG, Y. 2009. Impacts of topology and traffic pattern on capacity of hybrid wireless networks. *IEEE Trans. Mobile Comput. 8*, 12, 1585–1595

LI, X.-Y. 2009. Multicast capacity of wireless ad hoc networks. *IEEE/ACM Trans. Netw. 17*, 3, 950–961.

LI, X.-Y., TANG, S., AND FRIEDER, O. 2007. Multicast capacity for large scale wireless ad hoc networks. In *Proceedings of the 13th Annual International Conference on Mobile Computing and Networking (MobiCom'07)*. 266–277.

LI, X.-Y., ZHAO, J., WU, Y. W., TANG, S. J., XU, X. H., AND MAO, X. F. 2008. Broadcast capacity for wireless ad hoc networks. In *Proceedings of the 5th International Conference on Mobile Ad-hoc and Sensor Systems (MASS'08)*. 114–123.

LI, X.-Y., XU, X., WANG, S., TANG, S., DAI, G., ZHAO, J., AND QI, Y. 2009. Efficient data aggregation in multi-hop wireless sensor networks under physical interference model. In *Proceedings of the 6th International Conference on Mobile Ad-hoc and Sensor Systems (MASS'09)*. 353–362.

LI, X.-Y., LIU, Y., LI, S., AND TANG, S. 2010. Multicast capacity of wireless ad hoc networks under Gaussian channel model. *IEEE/ACM Trans. Netw. 18*, 4, 1145–1157.

LIU, B., TOWSLEY, D., AND SWAMI, A. 2008. Data gathering capacity of large scale multihop wireless networks. In *Proceedings of the 5th International Conference on Mobile Ad-hoc and Sensor Systems (MASS'08)*. 124–132.

LUO, C., WU, F., SUN, J., AND CHEN, C. W. 2009. Compressive data gathering for large-scale wireless sensor networks. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom'09)*. 145–156.

MAO, X., LI, X.-Y., AND TANG, S. 2008. Multicast capacity for hybrid wireless networks. In *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'08)*. 189–198.

MARCO, D., DUARTE-MELO, E. J., LIU, M., AND NEUHOFF, D. L. 2003. On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data. In *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN'03)*. Lecture Notes in Computer Science, vol. 2634, Spring-Verlag, Berlin Heidelberg, 1–16.

MOSCIBRODA, T. 2007. The worst-case capacity of wireless sensor networks. In *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN'07)*. 1–10.

NIESEN, U., GUPTA, P., AND SHAH, D. 2009. On capacity scaling in arbitrary wireless networks. *IEEE Trans. Inf. Theory 55*, 9, 3959–3982.

NIESEN, U., GUPTA, P., AND SHAH, D. 2010. The balanced unicast and multicast capacity regions of large wireless networks. *IEEE Trans. Inf. Theory 56*, 5, 2249–2271.

SHARMA, G., MAZUMDAR, R., AND SHROFF, N. B. 2007. Delay and capacity trade-offs in mobile ad hoc networks: A global perspective. *IEEE/ACM Trans. Netw. 15*, 5, 981–992

SIRKECI-MERGEN, B. AND GASTPAR, M. 2007. On the broadcast capacity of wireless networks. *IEEE Trans. Inf. Theory 56*, 8, 3847–3861.

TANG, S., LI, X.-Y., MAO, X., AND WANG, C. 2011. Impact of deployment size on the asymptotic capacity for wireless ad hoc networks under Gaussian channel model. *Wirel. Netw. 17*, 4, 817–832.

WAN, P.-J., HUANG, S. C.-H., WANG, L., WAN, Z., AND JIA, X. 2009. Minimum-latency aggregation scheduling in multihop wireless networks. In *Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'09)*. 185–194.

WANG, C., LIU, Y., LI, X.-Y., AND TANG, S. 2011. Aggregation capacity of wireless sensor networks: Extended network case. In *Proceedings of the 30th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'11)*. 1701–1709.

WANG, C., JIANG, C., LI, X.-Y., TANG, S., HE, Y., MAO, X., AND LIU, Y. 2012. Scaling laws of multicast capacity for power-constrained wireless networks under Gaussian channel model. *IEEE Trans. Comput. 61*, 5, 713–725.

WANG, X., BEI, Y., PENG, Q., AND FU, L. 2011. Speed improves delay-capacity trade-off in MotionCast. *IEEE Trans. Parall. Distrib. Syst. 22*, 5, 729–742.

WANG, Z., SADJADPOUR, H. R., AND GARCIA-LUNA-ACEVES, J. J. 2008. A unifying perspective on the capacity of wireless ad hoc networks. In *Proceedings of the 27th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'08)*. 211–215.

WANG, Z., SADJADPOUR, H. R., AND GARCIA-LUNA-ACEVES, J. J. 2008. The capacity and energy efficiency of wireless ad hoc networks with multi-packet reception. In *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'08)*. 179–188.

XU, Y. AND WANG, W. 2009. Scheduling partition for order optimal capacity in large-scale wireless networks. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom'09)*. 109–120.

YAN, M., HE, J. (S.), JI, S., AND LI, Y. 2011. Minimum latency scheduling for multi-regional query in wireless sensor networks. In *Proceedings of the 30th IEEE International Performance Computing and Communications Conference (IPCCC'11)*. 17–19.

ZHANG, G., XU, Y., WANG, X., AND GUIZANI, M. 2010. Capacity of hybrid wireless networks with directional antenna and delay constraint. *IEEE Trans. Commun. 58*, 7, 2097–2106.

ZHU, X., TANG, B., AND GUPTA, H. 2005. Delay efficient data gathering in sensor networks. In *Proceedings of the 1st International Converence on Mobile Ad-hoc and Sensor Networks (MSN'05)*. 380–389.

Received April 2012; revised July, September 2012; accepted September 2012