1.3.2.2 Lower Bound of Collection Capacity

The data collection algorithm based on branch scheduling in the BFS tree can still achieve the capacity of $\frac{W}{\tilde{\Delta}}$. However, in the general graph model, $\tilde{\Delta}$ is no longer bounded by a constant, and it could be $O(1)$ or $O(n)$. Thus, there is a gap between our lower bound of data collection $\frac{W}{\tilde{\Delta}}$ and the natural upper bound W. Considering both examples shown in Figure 1.12 of the article, the BFS tree-based method method matches their tight upper bounds $\frac{W}{n}$ and W. For the star topology, even though the sink has the maximal interference $\Delta = n$, each individual path has the path interference $\Delta_i = 1$, which leads to a capacity of W. For the straight-line topology, the path interference of the single path $\Delta_i = n$, thus the capacity is $\frac{W}{n}$. In both cases, $\frac{W}{\tilde{\Delta}}$ matches the optimal capacity. However, similar to $\frac{W}{\Delta}$, $\frac{W}{\tilde{\Delta}}$ is still not a tight bound. We will show such an example in Figure 1.14. In this subsection, we will provide two new tighter lower bounds for data collection in the general graph: one based on the branch scheduling method and the other based on a greedy scheduling method.

We first look at the branch scheduling–based method (Algorithm 1). We modify the basic path scheduling of the BFS tree-based method to achieve better collection capacity. Recall that in Section 1.3.1.2, we claim that the path scheduling for a path P_i can be done in $\Delta_i \times |P_i|$ time slots. However, we can perform path scheduling in the following way to save more time slots. Assume that path $P_i = s, v_1, v_2, \ldots v_{|P_i|}$ includes $|P_i|$ hops. Let $\delta_k^{P_i} = \max\left\{\delta^{P_i}(v_1), \ldots, \delta^{P_i}(v_k)\right\}$, that is, $\delta^{P_i}(v_k)$ is the maximum interference number among the first k nodes v_1 to v_k in path P_i. Clearly, $\delta^{P_i}(v_k) \le \delta^{P_i}(v_{k+1})$. In the first step, using $\delta_{|P_i|}^{P_i}$ slots, every node on the path transfers its data to its parent in the BFS tree. After the first step, the leaf $v_{|P_i|}$ already finishes its task in this round and has no data from the current snapshot. In the second step, using $\delta_{|P_i|-1}^{P_i}$ slots, the current snapshot data will move up one more level along the path in a BFS tree. Repeat these steps until all data along this path reaches the sink. It is easy to show that the total number of time slots used by the above procedure is $\sum_{k=1}^{|P_i|} \delta_k^{P_i}$. Because $\delta_k^{P_i} \le \Delta_i$, $\sum_{k=1}^{|P_i|} \delta_k^{P_i} \le \Delta_i \times |P_i|$.

Figure 1.14 shows an example in which $\sum_{k=1}^{|P_i|} \delta_k^{P_i}$ is much smaller than $\Delta_i \times |P_i|$. Again, we have n sensors and the sink distributed on a line P as shown in the figure. Assume that $R = r$. On the left side, there are $\log n$ nodes close to each other, thus their $\delta(v_i) = \log n$ except for $\delta(v_{n-\log n+1}) = \log n + 1$. On the right side, every node has $\delta(v_i) = 3$. Thus, $\Delta = \Delta_i = \log n + 1$ and $\Delta_i \times |P_i| = \Theta(n \log n)$. In addition, $\delta_k^P = \log n + 1$ for $k = n - \log n + 1, \ldots, n$ and $\delta_k^P = 3$ for $k = 3, \ldots, n - \log n$,

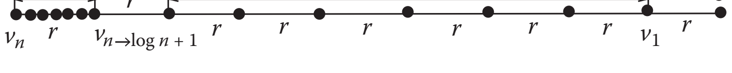

Figure 1.14 Illustration of the advantage of a new path scheduling. Here, *R* = *r*.

$\delta_2^P = 2$, and $\delta_1^P = 1$. Therefore, $\sum_{k=1}^{|P|} \delta_k^P = (\log n + 1)\log n + 3(n - \log n) - 3 = \Theta(n)$. It is obvious that $\sum_{k=1}^{|P|} \delta_k^P = \Theta(n)$ is smaller than $\Delta_i \times |P_i| = \Theta(n \log n)$ in order.

Using the new path scheduling analysis described above, we now derive a tight lower bound for our BFS tree-based method. Recall that our method transfers data based on branches in the BFS tree T. Given T, there are c paths P_i and c branches B_i as shown in Figure 1.10a and b. Then, the total number of time slots used by Algorithm 1 with the new path scheduling is at most

$$\sum_{i=1}^{c} \sum_{k=|P_i|-|B_i|+1}^{|P_i|} \delta_k^{P_i}.$$

It is clear that this number is much smaller than $\sum_{i=1}^{c} \left(\Delta_i \times |B_i|\right)$ from a previous analysis. Notice that for path P_i our algorithm (lines 3 and 4 in Algorithm 1) will terminate the transmission until branch B_i does not have data for the current snapshot and switches to the next path P_{i+1}. Thus, the index of k is only from $|P_i|$ to $|P_i| - |B_i| + 1$. Therefore, the capacity achieved by our algorithm is at least

$$\frac{W}{\dfrac{\sum_{i=1}^{c} \sum_{k=|P_i|-|B_i|+1}^{P_i} \delta_k^{P_i}}{n}}.$$

Let $\Delta^{**} = \dfrac{\sum_{i=1}^{c} \sum_{k=|P_i|-|B_i|+1}^{|P_i|} \delta_k^{P_i}}{n}$, which can be derived given the BFS tree. We now have a new lower bound of collection capacity as $\dfrac{W}{\Delta^{**}}$ [26,27]. Here, Δ^{**} is a kind of weighted average of the maximum interference among paths P_i and branches B_i in the BFS tree. We then have the following relationship:

$$n \geq \Delta \geq \tilde{\Delta} \geq \Delta^{**} \geq 1,$$

among the maximum interference number Δ in the whole graph, the maximum interference number $\tilde{\Delta}$ in the paths/branches of the BFS tree, and the "average" maximum interference Δ^{**} in the paths/branches of the BFS tree. These three interference numbers can be different from each other in order.

Now we introduce a new greedy-based scheduling algorithm inspired by Bonifaci et al. [32] and show that it can achieve a nice approximation ratio and lead to another tighter lower bound of collection capacity. The scheduling algorithm still uses the BFS tree as the collection tree. All messages will be sent along the branch toward the sink s. For n messages from one snapshot, it works as follows. In every time slot, it sends each message along the BFS tree from the current node to its parent, without creating interference with any higher-priority message. The priority ρ_i of each packet p_i is defined as $\frac{1}{l(v_i)}$. It is clear that packets originating from the children of the sink have the highest priority $\rho_i = 1$, whereas packets originating from other nodes have lower priority $\rho_i < 1$. For two packets with the same priority (on the same level in the BFS tree), ties can be broken arbitrarily. Given a schedule, let v_j^τ be the node of packet p_j in the end of time slot τ. The detailed greedy algorithm (Algorithm 2) is given in Figure 1.15.

Now we analyze the capacity achieved by this greedy data collection method. Before presenting the analysis, we first introduce some new notations. For two nodes v_i and v_j, $h(v_i, v_j)$ denotes the shortest hop number from v_i and v_j in graph G. The delay of packet p_j is defined as the time until it reaches the sink s, that is, $D_j = t \cdot \min\{\tau : v_j^\tau = s\}$.

Let λ_i be the minimal number of hops that a packet needs to be forwarded from node v_i before a new packet at v_i can be safely forwarded along the BFS tree. So $\lambda_i = \max\{l \mid \exists v_j, h(v_i, v_j) = l$ and transmission from v_i to $\text{par}(v_i)$ interferes with transmission from v_j to $\text{par}(v_j)\} + 1$. Here, $\text{par}(v_i)$ is the parent of v_i in T (see Figure 1.16 for illustration). Here, $\lambda_i = 4$ for v_i. We define $\lambda = \max_i\{\lambda_i\}$. Both λ and λ_i are integers (hop counts). In addition, we can prove $\lambda \geq \lambda^*$ as follows.

Algorithm 2 Greedy Scheduling on BFS Tree

Input: BFS tree T rooted at s.

Compute the priority $\rho_i = 1/l(v_i)$ of each message p_i.
for each snapshot **do**
  while $\exists p_j$ such that $\upsilon_j^\tau \neq s$ **do**
    for all such p_i in decreasing order of priority ρ_i **do**
      if sending p_i from node υ_j^τ will not create interference with any higher-priority messages that are already scheduled for this slot **then**
        node υ_j^τ sends p_i to its parent $\text{par}(\upsilon_j^\tau)$ in T.
      end if
    end for
    $\tau = \tau + 1$.
  end while
end for

Figure 1.15 Greedy scheduling on a BFS tree.

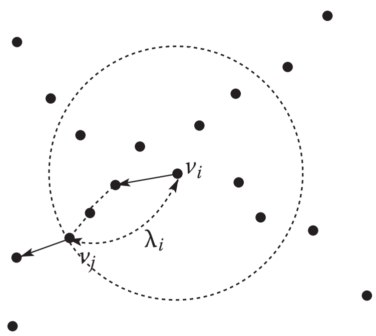

Figure 1.16 Illustration of the definitions of λ_i.

Lemma 8

In the general graph model, $\lambda \geq \lambda^*$ [27]. ■

Proof

Let v_k be the node inside critical region with the largest level. We now consider two cases.

Case 1: If there is a node outside the critical region, as shown in Figure 1.13a, the transmission from v_s to v_k should interfere with the transmission from v_q to s. Thus, in view of v_s, its $\lambda_s \geq l^* + 1 = \lambda^*$. Therefore, $\lambda \geq \lambda^*$.

Case 2: If all nodes are inside the critical region, again consider the v_k with the largest level. Then, $\lambda = \lambda_k = l(v_k) + 1 > l(v_k) = \lambda^*$.

Consequently, we conclude that $\lambda \geq \lambda^*$. ■

Packet p_j is said to be blocked in time slot τ if, in time slot τ, p_j is not sent out. We define the following blocking relation in our greedy algorithm schedule: $p_k \prec p_j$ if in the last time slot in which p_j is blocked by the transmission of higher priority packets in that time slot, p_k is the one closest to p_j in terms of hops among these packets (ties broken arbitrarily). The blocking relation induces a directed blocking tree T_D in which nodes are all message p_i and edge (p_k, p_j) representing $p_k \prec p_j$. The root p_r of the tree T_D is a message with the highest priority (originating from a child of s), which is never blocked. Let $P(j)$ the path in T_D from p_r to p_j and $h(j)$ be the hop count of $P(j)$. We then derive an upper bound on the delay D_j of packet p_j in the greedy algorithm.

Lemma 9

For each packet p_j in the snapshot [27], its delay

$$D_j \leq t \cdot \sum_{p_i \in P(j)} \min\{l(v_i), \lambda\}.$$

■

Proof

We prove this lemma by induction on $h(j)$. For any packet p_j, if $h(j) = 0$, which means p_j is the root p_r of T_D, it will not be blocked. So, $D_j \leq t \cdot l(v_j)$. Then, consider the right side of the inequation $t \cdot \sum_{p_i \in P(j)} \min\{l(v_i), \lambda\} = t \cdot \min\{l(v_j), \lambda\}$. Because p_j is the packet with the highest priority, $l(v_j) = 1$ and $l(v_j) \leq \lambda$. Thus, $t \cdot \sum_{p_i \in P(j)} \min\{l(v_i), \lambda\} = t \cdot l(v_j)$ and the claim in this lemma holds for the case in which $h(j) = 0$.

If $h(j) > 0$, that is, $p_j \neq p_r$, let τ be the last time slot in which p_j is blocked by packet p_k, that is, $p_k \prec p_j$. Notice that $t \cdot h\left(v_k^\tau, s\right) \leq D_k - t \cdot \tau$, otherwise p_k would not reach s by time D_k. Also, $h\left(v_j^\tau, v_k^\tau\right) \leq \lambda - 1$ because after p_k moves one hop, p_j is safe to move. From time slot $\tau + 1$, p_j may be forwarded toward s over one hop in each time slot, and reach s at the earliest time slot,

$$D_j \leq t \cdot \left[\tau + 1 + h\left(v_j^\tau, s\right)\right] \leq t \cdot \left[\tau + 1 + h\left(v_k^\tau, s\right) + h\left(v_j^\tau, v_k^\tau\right)\right]$$

$$\leq t \cdot (\tau + 1) + D_k - t \cdot \tau + t \cdot (\lambda - 1) = D_k + t \cdot \lambda.$$

On the other hand, $D_j \leq D_k + t \cdot l(v_j)$ because after p_k reaches the sink s, p_j needs at most $l(v_j)$ to reach the sink. Consequently, $D_j \leq D_k + t \cdot \min\{l(v_i), \lambda\}$. This completes our proof. ■

Lemma 10

The data collection capacity of our greedy algorithm [27] is at least $\frac{\lambda^*}{\lambda} \frac{W}{\Delta^*}$. ■

Proof

Let p_j be the packet having a maximum of D_j. By Lemma 9 and Lemma 8 ($\lambda \geq \lambda^*$),

$$D_j \leq t \sum_{p_i \in P(j)} \min\{l(v_i), \lambda\} \leq \frac{\lambda}{\lambda^*} t \sum_{p_i \in P(j)} \min\{l(v_i), \lambda^*\}$$

$$\leq \frac{\lambda}{\lambda^*} t \left[\sum_{v_i \in D(s,l^*)} l(v_i) + \sum_{v_i \notin D(s,l^*)} (l^* + 1)\right] = \frac{\lambda}{\lambda^*} t \sum_i \lambda_i^* = \frac{\lambda}{\lambda^*} nt\Delta^*$$

Thus, the capacity achieved by our greedy algorithm is at least $\frac{nb}{D_j} = \frac{\lambda^*}{\lambda} \frac{W}{\Delta^*}$. ■

In summary, we show that under the protocol and general graph models, the data collection capacity for arbitrary sensor networks has the following bounds:

Theorem 7

Under the protocol and general graph models [27], the data collection capacity for arbitrary sensor networks is at least $\frac{\lambda^*}{\lambda} \frac{W}{\Delta^*}$ and at most $\frac{W}{\Delta^*}$. ■

Here, λ^* describes the interference around the sink s, whereas λ describes the interference around a node v_i. Because $\lambda \geq \lambda^*$, $\frac{\lambda^*}{\lambda} \geq 1$. For the disk graph model, $\frac{\lambda^*}{\lambda}$ is a constant. However, for the general graph model, it may not be. Thus, there is still a gap between the lower and upper bounds (such an example is given in Figure 1.14). We leave finding tighter bounds to close the gap for future works. For two examples in Figure 1.12, the greedy method matches the optimal solutions in order. For the straight-line topology in Figure 1.12a, $\lambda = \lambda^* = n$ and $\Delta^* = \Theta(n)$. Thus, the capacity $\frac{\lambda^*}{\lambda}\frac{W}{\Delta^*} = \Theta\left(\frac{W}{n}\right)$ matches the upper bound. For the star topology in Figure 1.12b, $\lambda = \lambda^* = 1$ and $\Delta^* = 1$. In this case, $\frac{\lambda^*}{\lambda}\frac{W}{\Delta^*} = \Theta(n)$ also matches the upper bound. Compared with the branch scheduling method, the greedy method can achieve much better capacity in practice because it allows packet transmissions among multiple branches of the BFS tree in the same time slot.

Compared with the lower bound of $\frac{\lambda^*}{\lambda}\frac{W}{\Delta^*}$, which we derive from greedy scheduling on the BFS tree, this lower bound of $\frac{W}{\Delta^{**}}$, which we derive from branch scheduling on the BFS tree, may be smaller in some cases. Consider the example in Figure 1.14, $\frac{\lambda^*}{\lambda}\frac{W}{\Delta^*} = \Theta\left(\frac{W}{\log n}\right)$, whereas $\frac{W}{\Delta^{**}} = \frac{W}{\Delta^{**}} = \Theta(W)$. However, the reason is mainly due to the rough relaxation in our capacity analysis of greedy scheduling.

Finally, the bounds of collection capacity could be revised as the following:

Theorem 8

Under the protocol and general graph models [27], data collection capacity for arbitrary sensor networks is at least $\min\left\{\frac{\lambda^*}{\lambda}\frac{W}{\Delta^*}, \frac{W}{\Delta^{**}}\right\}$ and at most $\frac{W}{\Delta^*}$. ■

1.3.3 Data Collection under the Physical and Generalized Physical Models

Similar to the random network part, we can also consider data collection under a physical model or a generalized physical model instead of a protocol model for arbitrary networks.

1.3.3.1 Data Collection under the Physical Model

Chen et al. [27] proved the following theorem for data collection in arbitrary WSNs under the physical model.

Theorem 9

Under the physical and disk graph models [27], the data collection capacity for arbitrary WSNs is $\Theta(W)$. ■

The basic idea of their proof is as follows. To give an upper bound on the capacity of data collection, an artificial transmission range r_0 and an artificial interference range R_0 are defined, such that (1) the receiving node v_j of a sender v_i is within distance r_0, and (2) a transmitting node v_k

will cause interference at node v_j within distance R_0. That is, if there is any interference among the nodes in the protocol model with these artificial ranges, there is also interference among them in the physical model. By artificially setting r_0 and R_0 (which are both constants), we convert the physical model into a protocol model. Using previous proofs in protocol model, it is straightforward to show that the upper bound on the capacity under the disk graph model is bounded by $\Theta(W)$. Similarly, to give a lower bound on the capacity of data collection, an artificial transmission range r_1 and an artificial interference range R_1 are defined, such that, when all simultaneously transmitting nodes are separated by a distance R_1, and the receiving nodes of a transmitting node is within r_1, the SINR of every receiving node is at least η. In other words, if there is no interference among nodes in the protocol model with artificial ranges r_1 and R_1, there is no interference among the nodes in the physical model as well. Thus, we can convert the physical model into a protocol model. Using previous collection algorithms for the protocol model, it can be shown that the lower bound $\Theta(W)$ on the capacity of data collection under the disk graph model is achievable.

1.3.3.2 Data Collection under the Generalized Physical Model

For the capacity of data collection under a generalized physical model, we can derive an upper bound by considering the congestion near the sink node. In particular, we can prove that whatever scheduling scheme is implemented, the total transmission rate of all the incoming links at the sink node is upper bounded by some value. As a bottleneck, the capacity of the whole network is always bounded by that value, as stated in the following theorem.

Theorem 10

Under the generalized physical and general graph models [27], data collection capacity for arbitrary sensor networks is at most

$$\max_i(W_{is}) + W \cdot \log_2 n.$$

■

The first part of this upper bound depends on the rate of the shortest incoming link at the sink, whereas the second part depends on the total number of nodes. Notice that $\max_i(W_{is}) \le W \cdot \log_2\left(1+\frac{P}{N_0}\right)$. Thus, which part of the bound plays an important role depends on the relationship between n and $1+\frac{P}{N_0}$. When the network is a regular grid or a random homogeneous topology, we have $\max_i(W_{is}) + O(W \log n)$. Therefore, the total rate of all incoming links at sink node s is at most $O((\log n)W)$. The detailed proof of this theorem is similar to the one for Lemma 4 in Section 1.2.4.2, and it is true for any general graph. A lower bound of data collection capacity in this model is still open.

1.4 Conclusion

In this chapter, we investigate the theoretical limitations of data collection in terms of capacity for both random and arbitrary WSNs under different communication models. Table 1.1 briefly summarizes all completed work.

Table 1.1 Summary of Capacity Limits on Data Collection in WSNs

Network Model	*Graph Model*	*Communication Model*	*Sink no. k*	*Capacity C*
Random net	Disk graph	Protocol	1	$C = \Theta(W)$
Random net	Disk graph	Protocol	k (regularly deployed)	$C = \Theta(kW)$ if $k = O\left(\frac{n}{\log n}\right)$ $C = \Theta\left(\frac{n}{\log n}W\right)$ if $k = \Omega\left(\frac{n}{\log n}\right)$
Random net	Disk graph	Protocol	k (regularly deployed)	$\Theta\left(\frac{k}{\log k}W\right) \le C \le C = \Theta(kW)$ if $k = O\left(\frac{n}{\log n}\right)$ $C = \Theta\left(\frac{n}{\log n}W\right)$ if $k = \omega\left(\frac{n}{\log n}\right)$
Random net	Disk graph	Physical	1	$C = \Theta(W)$
Random net	Disk graph	Generalized physical	1	$\Omega\left((\log n)^{-\frac{p}{2}}\right) \le C \le O((\log n)W)$
Arbitrary net	Disk graph	Protocol	1	$C = \Theta(W)$
Arbitrary net	General graph	Protocol	1	$\min\left\{\Theta\left(\frac{\lambda^*}{\lambda}\frac{W}{\Delta^*}\right), \Theta\left(\frac{W}{\Delta^{**}}\right)\right\} \le C \le \Theta\left(\frac{W}{\Delta^*}\right)$
Arbitrary net	Disk graph	Physical	1	$C = \Theta(W)$
Arbitrary net	General graph	Generalized physical	1	$C \le \max_i(W_{i0}) + W \cdot \log_2(n)$

There are other advanced techniques, which can be applied in the data collection process to further improve capacity, such as using multiradios to reduce interference [18], using data aggregation to merge data packets [22,23,33], or using compressive data gathering to compress data packets [18,19]. For example, if each sensor can aggregate its received data (multiple packets) into a single packet, the following theorem can be proved, showing improved data rate and capacity for random networks over Theorem 1.

Theorem 11

Under the protocol model [22,23], the delay rate Γ and the capacity C of data aggregation in random sensor networks with a single sink are $\Theta\left(\sqrt{n \log n} W\right)$ and $\Theta\left(\frac{n}{\log n} W\right)$, respectively. ■

Notice that for data collection, the delay rate and the capacity are in the same order (Theorem 1), that is, pipelining can improve only a constant factor of the data rate. However, for data aggregation, it is very interesting to see that pipelining can increase the data rate in order.

Finally, all results presented in this chapter focus on how fast the data collection can be performed under the existence of interferences among sensors. However, in practice, there are also other metrics that should be considered for data collection in WSNs, such as total energy consumption [33], message complexity [33], load balancing among sensors, or possible retransmissions [21]. Readers are encouraged to check relevant references in the literature.

Acknowledgments

The author is grateful to his PhD students (Siyuan Chen and Minsu Huang) and colleagues (Xiang-Yang Li, Shaojie Tang, and Xinghua Shi) for working together on this research topic. This work was supported in part by the U.S. National Science Foundation under grants CNS-0915331 and CNS-1050398.

References

1. I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine* 40:102–114, 2002.
2. C. Wang, C. Jiang, X.-Y. Li, S. Tang, Y. He, X. Mao, and Y. Liu. Scaling laws of multicast capacity for power-constrained wireless networks under Gaussian channel model. *IEEE Transactions on Computers* 61(5):713–725, 2012.
3. A. Agarwal and P. R. Kumar. Capacity bounds for ad hoc and hybrid wireless networks. *ACM SIGCOMM Computer Communication Review* 34(3):71–81, 2004.
4. P. Gupta and P. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory* 46(2):388–404, 2000.
5. M. Grossglauser and D. Tse. Mobility increases the capacity of ad hoc wireless networks. In *Proceedings of IEEE INFOCOM*, 2001.

6. B. Liu, P. Thiran, and D. Towsley. Capacity of a wireless ad hoc network with infrastructure. In *Proceedings of ACM MobiHoc '07*, 2007.
7. X.-Y. Li, S. Tang, and O. Frieder. Multicast capacity for large scale wireless ad hoc networks. In *Proceedings of ACM MobiCom*, 2007.
8. X. Mao, X.-Y. Li, and S. Tang. Multicast capacity for hybrid wireless networks. In *Proceedings of ACM MobiHoc '08*, 2008.
9. S. Shakkottai, X. Liu, and R. Srikant. The multicast capacity of large multihop wireless networks. In *Proceedings of ACM MobiHoc '07*, 2007.
10. A. Keshavarz-Haddad, V. Ribeiro, and R. Riedi. Broadcast capacity in multihop wireless networks. In *Proceedings of ACM MobiCom*, 2006.
11. B. Tavli. Broadcast capacity of wireless networks. *IEEE Communications Letters* 10:68–69, 2006.
12. E. J. Duarte-Melo and M. Liu. Data-gathering wireless sensor networks: Organization and capacity. *Computer Networks* 43:519–537, 2003.
13. M. Liu, D. L. Neuhoff, D. Marco, and E. J. Duarte-Melo. On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data. In *Proceedings of ACM IPSN*, 2003.
14. H. El Gamal. On the scaling laws of dense wireless sensor networks: The data gathering channel. *IEEE Transactions on Information Theory* 51(3):1229–1234, 2005.
15. R. J. Barton and R. Zheng. Order-optimal data aggregation in wireless sensor networks using cooperative time-reversal communication. In *Proceedings of the Annual Conference on Information Sciences and Systems*, 2006.
16. R. Zheng and R. J. Barton. Toward optimal data aggregation in random wireless sensor networks. In *Proceedings of IEEE INFOCOM*, 2007.
17. B. Liu, D. Towsley, and A. Swami. Data gathering capacity of large scale multihop wireless networks. In *Proceedings of IEEE MASS*, 2008.
18. S. Ji, Y. Li, and X. Jia. Capacity of dual-radio multi-channel wireless sensor networks for continuous data collection. In *Proceedings of IEEE INFOCOM*, 2011.
19. S. Ji, R. Beyah, and Y. Li. Continuous data collection capacity of wireless sensor networks under physical interference model. In *Proceedings of IEEE MASS 2011*, 2011.
20. S. Ji and Z. Cai. Distributed data collection and its capacity in asynchronous wireless sensor networks. In *Proceedings of IEEE INFOCOM*, 2012.
21. S. Ji, R. Beyah, and Z. Cai. Snapshot/continuous data collection capacity for large-scale probabilistic wireless sensor networks. In *Proceedings of IEEE INFOCOM*, 2012.
22. S. Chen, Y. Wang, X.-Y. Li, and X. Shi. Order-optimal data collection in wireless sensor networks: Delay and capacity. In *Proceedings of 6th Annual IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON 2009)*, 2009.
23. S. Chen, Y. Wang, X.-Y. Li, and X. Shi. Capacity of data collection in randomly-deployed wireless sensor networks. *ACM Springer Wireless Networks* 17(2):305–318, 2011.
24. S. Chen, Y. Wang, X.-Y. Li, and X. Shi. Data collection capacity of random-deployed wireless sensor networks. In *Proceedings of IEEE Global Telecommunications Conference (Globecom 2009)*, 2009.
25. S. Chen and Y. Wang. Data collection capacity of random-deployed wireless sensor networks under physical models. *Tsinghua Science and Technology* 17(5):485–498, 2012.
26. S. Chen, S. Tang, M. Huang, and Y. Wang. Capacity of data collection in arbitrary wireless sensor networks. In *Proceedings of IEEE 29th Conference on Computer Communications (INFOCOM 2010), Mini-Conference*, 2010.
27. S. Chen, M. Huang, S. Tang, and Y. Wang. Capacity of data collection in arbitrary wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems* 23(1):52–60, 2012.
28. C. Wang, X.-Y. Li, C. Jiang, and S. Tang. General capacity scaling of wireless networks. In *Proceedings of IEEE INFOCOM*, 2011.
29. S. R. Kulkarni and P. Viswanath. A deterministic approach to throughput scaling in wireless networks. *IEEE Transactions on Information Theory* 50(6):1041–1049, 2004.
30. S. Rao. The *m* balls and *n* bins problem. Lecture Note for Lecture 11, CS270, University of California, Berkeley, 2003.