IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS VOL.24 NO.1 YEAR 2012

Capacity of Data Collection in Arbitrary Wireless Sensor Networks

Siyuan Chen* Shaojie Tang† Minsu Huang* Yu Wang*
*Department of Computer Science, University of North Carolina at Charlotte, Charlotte, North Carolina, USA
†Department of Computer Science, Illinois Institute of Technology, Chicago, Illinois, USA

***Abstract*— How to efficiently collect sensing data from all sensor nodes is critical to the performance of wireless sensor networks. In this paper, we aim to understand the theoretical limitations of data collection in terms of possible and achievable maximum capacity. Previously, the study of data collection capacity [1]–[6] has only concentrated on large-scale *random* networks. However, in most of practical sensor applications, the sensor network is not deployed uniformly and the number of sensors may not be as huge as in theory. Therefore, it is necessary to study the capacity of data collection in an arbitrary network. In this paper, we derive the upper and constructive lower bounds for data collection capacity in arbitrary networks. The proposed data collection method can lead to order-optimal performance for any arbitrary sensor networks. We also examine the design of data collection under a general graph model and discuss performance implications.**

I. Introduction

A wireless sensor network consists of a set of sensor devices which spread over a geographical area. The ultimate goal of sensor networks is often to collect the sensing data from all sensors to a sink node and then perform further analysis at the sink node. In this paper, we study some fundamental capacity problems arising from data collection scenario in wireless sensor networks. We consider a wireless sensor network where n sensors are *arbitrarily* deployed in a finite geographical region. Each sensor measures independent field values at regular time intervals and sends these values to a sink node. The union of all sensing values from n sensors at a particular time is called *snapshot*. The task of data collection is to deliver these snapshots to a single sink. Due to spatial separation, several sensors can successfully transmit at the same time if these transmissions do not cause any destructive wireless interferences. As in the literature, the classical *protocol interference model* is used in our analysis, while all analysis results can also be extended to *physical interference model* by applying the technique introduced in [7]. We also assume that a successful transmission over a link has a fixed data-rate W bit/second.

The performance of data collection in sensor networks can be characterized by the rate at which sensing data can be collected and transmitted to the sink node. In particular, the theoretical measure that captures the limitations of collection processing in sensor networks is capacity for the many-to-one data collection, *i.e.*, the maximum data rate at the sink to continuously receive the snapshot data from sensors. *Data collection capacity* reflects how fast the sink can collect sensing data from all sensors under existence of interference. It is critical to understand the limitation of many-to-one information flows and devise efficient data collection algorithms to maximize the performance of wireless sensor networks.

Capacity limits of data collection in random wireless sensor networks have been studied in the literature [1]–[6]. In [1], [2], Duarte-Melo *et al.* first introduced the many-to-one transport capacity in dense and random sensor networks under protocol interference model. El Gamal [3] studied the capacity of data collection subject to a total average transmitting power constraint where a node can receive data from multiple source nodes at a time. Barton and Rong [4] also investigated the capacity of data collection under general physical layer models (e.g. cooperative time reversal communication model) where the data rate of individual link is not fixed as a constant W but various depending on the transmitting powers and transmitting distances of all simultaneous transmissions. Both [3] and [4] adopted complex physical layer techniques, such as antenna sharing, channel coding and cooperative beamforming, in their models. Liu *et al.* [5] recently studied the capacity of a general some-to-some communication paradigm under protocol interference model in random networks where there are multiple randomly selected sources and destinations. They derived the upper and constructive lower bounds for such a problem. Chen *et al.* [6] studied the capacity of data collection under protocol interference model with multiple sinks. However, all the above research shares the common assumption where large number of sensor nodes are either located on a grid structure or randomly and uniformly distributed in a plane. Such assumption is useful for simplifying the analysis and deriving nice theoretical limitations, but may be invalid in many practical sensor applications. To our best knowledge, our paper is the first one to study data collection capacity for arbitrary networks.

In this paper, we focus on *deriving capacity bounds of data collection for arbitrary networks*, where sensor nodes are deployed in any distribution and can form any network topology. We summarize our contributions as follows:

- For arbitrary sensor network under protocol interference model, we propose a data collection method based on Breadth First Search (BFS) tree. We prove that this method can achieve collection capacity of $\Theta(W)$ which matches the theoretical upper bound.

This work is supported in part by the US National Science Foundation (NSF) under Grant No. CNS-0721666 and No. CNS-0915331.

- Since disk graph model is idealistic, we also consider a more practical model: *general graph model*. In general graph model, two nearby nodes may be unable to communicate due to various reasons such as barrier and path fading. We first show that $\Theta(W)$ may not be achievable for a general graph. Then we prove that BFS-based method can still achieve capacity of $\Theta(\frac{W}{\Delta^*})$ where Δ^* is a new interference parameter defined in Section IV.

The results above not only help us to understand the theoretical limitations of data collection in sensor networks, but also provide practical and efficient data collection methods (including how to construct data collection structure and how to schedule data collection) to achieve near-optimal capacity (within constant times of the optimal). Even though we are focusing on arbitrary networks, all of our solutions can be applied to random networks since any random network is just a special case of arbitrary networks.

II. Network Models and Collection Capacity

A. Basic Network Models

In this paper, we focus on the capacity bound of data collection in arbitrary wireless sensor networks. For simplicity, we start with a set of simple and yet general enough models that are widely used in the community.

We consider an arbitrary wireless network with n sensor nodes $v_1, v_2, \cdots, v_n$ and a single sink v_0. These n sensors are arbitrarily distributed in a field. At regular time intervals, each sensor measures the field value at its position and transmits the value to the sink. We adopt a *fixed data-rate channel model* where each wireless node can transmit at W bits/second over a common wireless channel. We also assume that all packets have unit size b bits. The time is divided into time slots with $t = b/W$ seconds. Thus, only one packet can be transmitted in a time slot between two neighboring nodes. TDMA scheduling is used at MAC layer.

Under the fixed data-rate channel model, we assume that every node has a fixed transmission power P. Thus, a fixed transmission range r can be defined such that a node v_j can successfully receive the signal sent by node v_i only if $||v_i - v_j|| \leq r$. Here, $||v_i - v_j||$ is the Euclidean distance between v_i and v_j. We call this model *disk graph model*. We can further define a communication graph $G = (V, E)$ where V is the set of all nodes (including the sink) and E is the set of all possible communication links. In this paper, we always assume graph G is connected.

Due to spatial separation, several sensors can successfully transmit at the same time if their transmissions do not cause any destructive wireless interferences. As in the literature, we model the interference using *protocol interference model*. All nodes have a uniform interference range R. When node v_i transmits to node v_j, node v_j can receive the signal successfully if no node within a distance R from v_j is transmitting simultaneously. Here, for simplicity, we assume that $\frac{R}{r}$ is a constant α which is larger than 1. Let $\delta(v_i)$ be the number of nodes in v_i's interference range (including v_i itself) and Δ be the maximum value of $\delta(v_i)$ for all nodes v_i, $i = 0, \cdots, n$.

B. Capacity of Data Collection

We now formally define delay and capacity of data collection in wireless sensor networks. Recall that each sensor generates a field value with b bits at regular time intervals, and tries to transport it to the sink. We call the union of all values from all n sensors at particular sampling time a *snapshot* of the sensing data. Then the goal of data collection is to collect these snapshots from all sensors. It is clear that the sink prefer to get each snapshot as quickly as possible. In this paper, we assume that there is no correlation among all sensing values and no network coding or aggregation technique is used during the data collection.

Definition 1: The **delay of data collection** D is the time used by the sink to successfully receive a snapshot, i.e., the time needed between completely receiving one snapshot and completely receiving the next snapshot at the sink.

Definition 2: The **capacity of data collection** C is the ratio between the size of data in one snapshot and the time to receive such a snapshot (i.e., $\frac{nb}{D}$) at the sink.

Thus, the capacity C is the maximum data rate at the sink to continuously receive the snapshot data from sensors. Here, we require the sink to receive the complete snapshot from all sensors (*i.e.*, data from all sensors need to be delivered). Notice that data transport can be pipelined in the sense that further snapshots may begin to transport before the sinks receiving prior snapshots. In this paper, we focus on *capacity analysis of data collection in an arbitrary sensor network.*

III. Collection Capacity under Disk Graph Model

Upper Bound of Collection Capacity: It has been proved that the upper bound of capacity of data collection for random networks is W [1], [2]. It is obviously that this upper bound also holds for any arbitrary network. The sink v_0 cannot receive at rate faster than W since W is the fixed transmission rate of individual link. Therefore, we are interested in design of data collection algorithm to achieve capacity in the same order of the upper bound, *i.e.* $\Theta(W)$.

We now propose a BFS-based data collection method and demonstrate that it can achieve the capacity of $\Theta(W)$ under our network model. Our data collection method includes two steps: data collection tree formation and data collection schedulling.

A. Data Collection Tree - BFS Tree

The data collection tree used by our method is a classical Breadth First Search (BFS) tree rooted at the sink v_0. The time complexity to construct such a BFS tree is $O(|V|+|E|)$. Let T be the BFS tree and $v_1^l, \cdots, v_m^l$ be all leaves in T. For each leaf v_i^l, there is a path P_i from itself to the root v_0. Let $\delta^{P_i}(v_j)$ be the number of nodes on path P_i which are inside the interference range of v_j (including v_j itself). Assume the maximum interference Δ_i on each path P_i is $\max\{\delta^{P_i}(v_j)\}$ for all $v_j \in P_i$. Hereafter, we call Δ_i *path interference* of path P_i. Then we can prove that T has a nice property that the path interference of each branch is bounded by a constant.

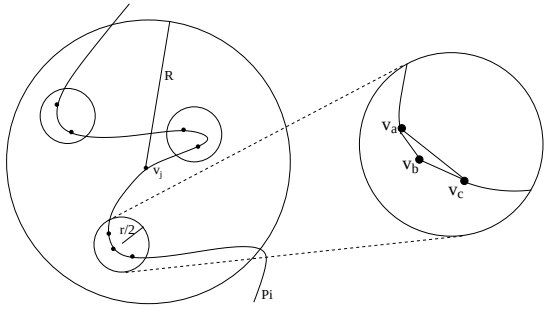


Fig. 1. Proof of Lemma 1: on a path P_i in BFS T, the interference nodes for a node v_j is bounded by a constant.

Lemma 1: Given a BFS tree T under the protocol interference model, the maximum interference Δ_i on each path P_i is bounded by a constant $8\alpha^2$, i.e., $\Delta_i \leq 8\alpha^2$.

Proof: We prove by contradiction with a simple area argument. Assume that there is a v_j on P_i whose $\delta^{P_i}(v_j) > 8\alpha^2$. In other words, more than $8\alpha^2$ nodes on P_i are located in the interference region of v_j. Since the area of interference region is πR^2, we consider the number of interference nodes inside a small disk with radius $\frac{r}{2}$. See Figure 1 for illustration. The number of such small disks is at most $\frac{\pi R^2}{\pi(\frac{r}{2})^2} = 4\alpha^2$ inside πR^2. By the Pigeonhole principle, there must be more than $\frac{8\alpha^2}{4\alpha^2} = 2$ nodes inside a single small disk with radius $\frac{r}{2}$. In other words, three nodes v_a, v_b and v_c on the path P_i are connected to each other as shown in Figure 1. This is a contradiction with the construction of BFS tree, since one of such nodes will be visited on other path (*i.e.* on a single path a node can only connected to two other nodes (its parent and child on the path)). As shown in Figure 1, if v_a and v_c are connected in G, then v_c should be visited on the other path instead of P_i. This finishes our proof. ■

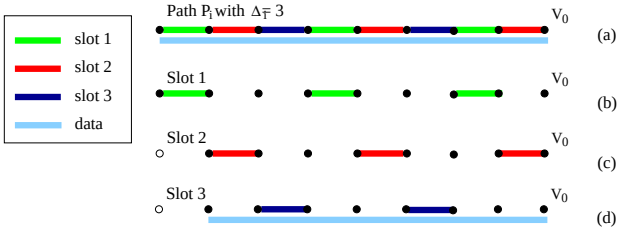


Fig. 2. Scheduling on a path: after Δ_i slots the sink gets one data.

B. Scheduling Algorithm

We now illustrate how to collect one snapshot from all sensors. Given the collection tree T, our scheduling algorithm basically collects data from each path P_i in T one by one.

First, we explain how to schedule collection on a single path. For a given path P_i, we can use Δ_i slots to collect one data in the snapshot at the sink. See Figure 2 for illustration. In this figure, we assume that $R = r$ and only adjacent nodes interfere with each other. Thus $\Delta_i = 3$. Then we color the path using green, red, and blue as in Figure 2(a). Every node on the path has unit data to transfer. Green, red and blue links are active in the first slot, the second slot and the third slot, respectively. After three slots (Figure 2(d)), the leaf node has no data in this snapshot and the sink gets one data from its child. Therefore, to receive all data on the path, at most $\Delta_i \times |P_i|$ time slots are needed. We call this scheduling method *Path Scheduling*.

Now we describe our scheduling algorithm on the collection tree T. Remember T has m leaves which define m pathes from P_1 to P_m. Our algorithm collects data from path P_1 to P_m in order. We define that i-th branch B_i is the part of P_i from v_i^l to the intersection node with P_{i+1} for $i = [1, m-1]$ and m-th branch $B_m = P_m$. For example, in Figure 3(b), there are four branches in T: B_1 is from v_1^l to v_a, B_2 is from v_2^l to v_0, B_3 is from v_3^l to v_b, and B_4 is from v_4^l to v_0. Remember that the union of all branches is the whole tree T. Algorithm 1 shows the detailed scheduling algorithm.

Algorithm 1 Data Collection Scheduling on BFS

Input: BFS tree T.

1: **for** each snapshot **do**
2: **for** $t = 1$ to m **do**
3: Collect data on path P_i. All nodes on P_i transmit data towards the sink v_0 using *Path Scheduling*.
4: The collection terminates when nodes on branch B_i do not have data for this snapshot. Notice that the total slots used are at most $\Delta_i \cdot |B_i|$, where $|B_i|$ is the hop length of B_i.

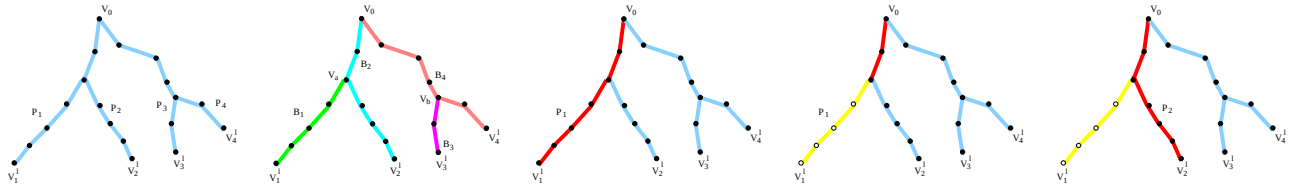
Figure 3(c)-(j) give an example of scheduling on T. In the first step (Figure 3(c)), all nodes on P_1 participate in the transmission using the scheduling method for a single path (every Δ_1 slots, sink v_0 receives one data). Such transmission stops until there is no data in this snapshot on branch B_1, as shown in Figure 3(d). Then in the second step data on path P_2 is transmitted. This procedure repeats until all data in this snapshot reach v_0.

C. Capacity Analysis

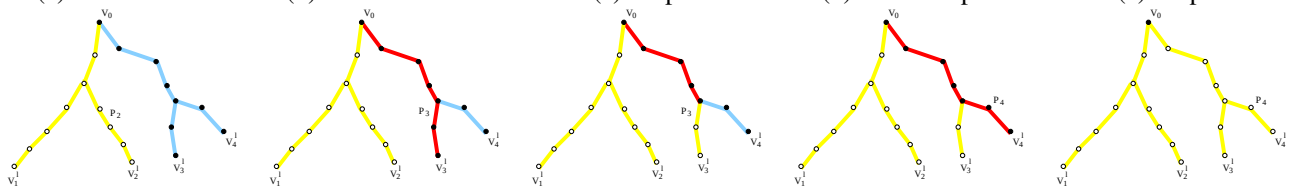
We now analyze the achievable capacity of our data collection method by counting how many time slots the sink needs to receive all data in one snapshot.

Theorem 2: The BFS-based data collection method can achieve data collection capacity of $\Theta(W)$ at the sink.

Proof: In Algorithm 1, the sink collects data from all m pathes in T. In each step (Lines 3-4), data are transferred on path P_i and it takes at most $\Delta_i \cdot |B_i|$ time slots. Recall that *Path Scheduling* needs at most $\Delta_i \cdot k$ time slots to collect k packets from path P_i. Therefore, the total number of time slots needed for Algorithm 1, denoted by τ, is at most $\sum_{i=1}^{m} \Delta_i \cdot |B_i|$. Since the union of all branches is the whole tree T, *i.e.*, $\sum_{i=1}^{m} |B_i| = n$. Thus, $\tau \leq \sum_{i=1}^{m} \Delta_i |B_i| \leq \sum_{i=1}^{m} \tilde{\Delta} |B_i| \leq \tilde{\Delta} n$. Here $\tilde{\Delta} = \max\{\Delta_1, \cdots, \Delta_m\}$. Then, the delay of data collection $D = \tau t \leq \tilde{\Delta} n t$. The capacity $C = \frac{nb}{D} \geq \frac{nb}{\tilde{\Delta} n t} = \frac{W}{\tilde{\Delta}}$. From Lemma 1, we know that $\tilde{\Delta}$ is bounded by a constant. Therefore, the data collection capacity is $\Theta(W)$. ■



(a) BFS Tree T (b) Branches in T (c) Step 1 (d) After Step 1 (e) Step 2



(f) After Step 2 (g) Step 3 (h) After Step 3 (i) Step 4 (j) After Step 4

Fig. 3. Illustrations of our scheduling on the data collection tree T.

Recall that the upper bound of data collection capacity is W, thus our data collection algorithm is order-optimal. Consequently, we have the following theorem.

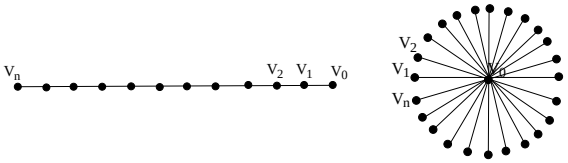
Theorem 3: Under protocol interference model and disk graph model, data collection capacity for arbitrary wireless sensor networks is $\Theta(W)$.

IV. Collection Capacity under General Graph

So far, we assume that the communication graph is a disk graph where two nodes can communicate if and only if their distance is less than or equal to transmission range r. However, a disk graph model is idealistic since in practice two nearby nodes may be unable to communicate due to various reasons such as barrier and path fading. Therefore, in this section, we consider a new general graph model $G = (V, E)$ where V is the set of sensors and E is the set of possible communication links. Every sensor still has a fixed transmission range r such that the necessary condition for v_j to receive correctly the signal from v_i is $||v_i - v_j|| \leq r$. Notice that $||v_i - v_j|| \leq r$ is not the sufficient condition for an edge $v_iv_j \in E$. Some links do not belong to G because of physical barriers or the selection of routing protocols. Thus, G is a subgraph of a disk graph. Under this model, the network topology G can still be any general graph (for example, setting $r = \infty$ and putting a barrier between any two nodes v_i and v_j if $v_iv_j \notin G$).

A. Data Collection under General Graph Model

In the new general graph model, the capacity of data collection could be $\frac{W}{n}$ in the worst-case. We consider a simple straight-line network topology with n sensors as shown in Figure 4(a). Assume that the sink v_0 is located at the end of the network and the interference range is large enough to cover every node in the network. Since transmission on one link will interfere with all other nodes, the only possible scheduling is transferring data along the straight-line via all links. The total time slots needed are $n(n+1)/2$, thus the capacity is at most $\Theta(\frac{W}{n})$. Notice that in this example, the maximum interference Δ of graph G is n. It seems the upper bound of data collection capacity could be $\frac{W}{\Delta}$. We now show an example whose capacity can be much larger than $\frac{W}{\Delta}$. Again we assume all n nodes with the sink interfering with each other. The network topology is a star with the sink v_0 in center, as shown in Figure 4(b). Clearly, a scheduling which lets every node transfer data in order can lead to a capacity W which is much larger than $\frac{W}{\Delta} = \frac{W}{n}$.



(a) Straight-line Topology (b) Star Topology

Fig. 4. The optimum of BFS-based method under two extreme cases.

Fortunately, the BFS-based data collection algorithm still works well under general graph model. It is easy to see that the capacity is still $\frac{W}{\tilde{\Delta}}$. Here, $\tilde{\Delta}$ is the maximum path interference among all pathes. However, in general graph model we can not bound $\tilde{\Delta}$ by a constant any more, 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 4, the BFS-based 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 capacity of $\frac{W}{1} = W$. For the straight-line topology, the path interference of the single path $\Delta_i = n$, thus the capacity is $\frac{W}{n}$.

B. Tighter Lower Bound

Actually $\frac{W}{\tilde{\Delta}}$ is not a tight lower bound by BFS-based method. Now we are ready to show a tighter lower bound by reconsidering how to do the *Path Scheduling*. In Section III we claimed that the path scheduling for a path P_i can be done in $\Delta_i \cdot |P_i|$ time slots. However, we can perform path scheduling in the following way to save more slots. Assume that path $P_i = v_0, v_1, v_2, \cdots, v_{|P_i|}$. Let $\delta_k^{P_i} = \max\{\delta^{P_i}(v_1), \cdots, \delta^{P_i}(v_k)\}$,

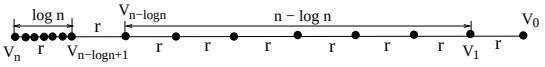


Fig. 5. Illustration of the advantage of a new path scheduling.

i.e., $\delta_k^{P_i}$ is the maximum interference among *first* k nodes v_1 to v_k in path P_i. In the first step, using $\delta_{|P_i|}^{P_i}$ slots, every node on the path transfers its data to its parent. After the first step, the leaf $v_{|P_i|}$ already finishes its task in this round and has no data from current snapshot. In the second step, using $\delta_{|P_i|-1}^{P_i}$ slots, the current snapshot data will move one more level up along the path in the BFS tree. Repeat these steps until all data along this path reach the sink. It is easy to show that the total number of time slots used by the above procedure is $\sum_{k=1}^{|P_i|} \delta_k^{P_i}$. Since $\delta_k^{P_i} \leq \Delta_i$, $\sum_{k=1}^{|P_i|} \delta_k^{P_i} \leq \Delta_i \cdot |P_i|$. Figure 5 shows an example where $\sum_{k=1}^{|P_i|} \delta_k^{P_i}$ is much smaller than $\Delta_i \cdot |P_i|$. 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, $\Delta = \log n + 1$ and $\Delta \cdot |P| = \Theta(n \log n)$. In addition, $\delta_k^P = \log n + 1$ for $k = n - \log n + 1, \cdots, n$ and $\delta_k^P = 3$ for $k = 3, \cdots, n - \log n$, $\delta_2^P = 2$, 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 $\Delta \cdot |P| = \Theta(n \log n)$ in order.

Based on the new path scheduling analysis, we now derive a tighter lower bound for our BFS-based method. Recall that our method transfers data based on branches in BFS tree T. In T, there are m pathes P_i and m branches B_i as shown in Figure 3(a) and 3(b). Then the total number of time slots used by Algorithm 1 with new path scheduling is at most

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

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

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

Let $\Delta^* = \frac{\sum_{i=1}^{m} \sum_{k=|P_i|-|B_i|+1}^{|P_i|} \delta_k^{P_i}}{n}$ which can be derived given the BFS tree. Here Δ^* is a kind of weighted-average of the maximum interference among pathes 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 Δ in the whole graph, the maximum interference $\tilde{\Delta}$ in the pathes/braches of the BFS tree, and the "average" maximum interference Δ^* in the pathes/branches of the BFS tree. These three interference numbers can be different from each other in order. Even though $\frac{W}{\Delta^*}$ is a tighter lower bound for data collection, there is still a gap between it and the upper bound W. Thus, we leave finding a tighter bound to close the gap as one of our future work.

Theorem 4: Under protocol interference model and general graph model, data collection capacity for arbitrary sensor networks is at least $\frac{W}{\Delta^*}$ and at most W.

V. Conclusion

In this paper, we study the theoretical limitations of data collection in terms of capacity for arbitrary wireless sensor networks. We first propose an efficient data collection method to achieve capacity of $\Theta(W)$, which is order-optimal under protocol interference model. However, when the underlying network model is a general graph, we show that $\Theta(W)$ may not be achievable. We prove that BFS-based method can still achieve capacity of $\Theta(\frac{W}{\Delta^*})$ for general graphs. All of our methods can also achieve these results for random networks.

There are still several open problems left as our future work. (1) We would like to close the gap of upper and lower bounds of data collection capacity for general graphs. (2) Even though the capacity of data aggregation for arbitrary networks has been studied in [8], they only consider the worst case capacity. It is interesting to study aggregation capacity for any arbitrary network. (3) Here we focus on achieving order-optimal capacity (*i.e.*, constant approximation for minimizing delay and maximizing capacity), but how to achieve optimal (or near-optimal) capacity (*i.e.*, reduce the approximation ratio) is a more challenging task. We leave it as one of our future work. Recall that some of the problems (*e.g.* minimum delay data aggregation [9]) are NP-hard.

References

[1] E.J. Duarte-Melo and M. Liu, "Data-gathering wireless sensor networks: Organization and capacity," *Computer Networks*, 43, 519-537, 2003.

[2] D. Marco, E.J. Duarte-Melo, M. Liu, and D.L. Neuhoff, "On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data," in *Proc. Int'l Workshop on Information Processing in Sensor Networks*, 2003.

[3] H.E. Gamal, "On the scaling laws of dense wireless sensor networks: the data gathering channel," *IEEE Trans. on I.T.*, 51(3):1229-1234, 2005.

[4] R. Zheng and R.J. Barton, "Toward optimal data aggregation in random wireless sensor networks," in *Proc. of IEEE Infocom*, 2007.

[5] B. Liu, D. Towsley, and A. Swami, "Data gathering capacity of large scale multihop wireless networks," in *Proc. of IEEE MASS*, 2008.

[6] S. Chen, Y. Wang, X.-Y. Li, X. Shi, "Order-optimal data collection in wireless sensor networks: Delay and capacity," in *IEEE SECON*, 2009.

[7] X.-Y. Li, J. Zhao, Y.W. Wu, S.J. Tang, X.H. Xu, X.F. Mao, "Broadcast capacity for wireless ad hoc networks," in *IEEE MASS*, 2008.

[8] T. Moscibroda, "The worst-case capacity of wireless sensor networks," in *Proc. of ACM IPSN*, 2007.

[9] S.C.-H. Huang, P.-J. Wan, C.T. Vu, Y. Li, and F. Yao, "Nearly constant approximation for data aggregation scheduling in wireless sensor networks," in *Proc. of IEEE INFOCOM*, 2007.