$$\frac{L}{d} = \left[\frac{1}{8\phi}\cdot\left(\frac{(\sqrt{5}d)^{-\beta}}{\eta d^{-\beta}} - \frac{N_0}{Pd^{-\beta}}\right)\right]^{-\frac{1}{\beta}} + 2 = \left[\frac{1}{8\phi}\cdot\left(\frac{5^{-\beta/2}}{\eta} - \frac{N_0 d^{\beta}}{P}\right)\right]^{-\frac{1}{\beta}} + 2.$$

When $n \to \infty$, this ratio goes to a constant, denoted by α.

Notice that now the physical model can be treated as a protocol model because the ratio between the sizes of interference block and micro cell $\frac{L}{d}$ is bounded by the constant α. By having interference blocks, we can simply use the data collection scheme for protocol model, which has already been presented in Section 1.2.2, to perform the data collection. It is easy to show that the same capacity can be achieved compared with the protocol model. In summary, we have the following theorem for the physical model:

Theorem 4

Under the physical model [24,25], the delay rate Γ and the capacity C of data collection in random sensor networks with a single sink are both $\Theta(W)$. ■

1.2.4.2 Data Collection under the Generalized Physical Model

The physical model assumes a threshold-based channel in which the signal can be decoded at a fixed constant rate of W bits per second only if the SINR is greater than a certain threshold. If the SINR is below this threshold, no throughput is received at all. However, in practice, the throughput is usually a function of the SINR at the receiver. Thus, the generalized physical model is a more realistic communication model than the protocol or physical models, especially under random extended networks [28]. Therefore, we also study the theoretical bounds of data collection capacity under the generalized physical model. Notice that because the data rate is now related to SINR and interference, the capacity analysis becomes much more complex and challenging.

First, we give a lemma to derive an upper bound of data collection capacity under the generalized physical model.

Lemma 4

Under the generalized physical model [25], the capacity of data collection in random sensor networks is at most $O[(\log n) W]$. ■

Proof

We first order all the incoming links of sink s according to their length as follows: $\|v_1 - s\| \le \|v_2 - s\| \le \ldots \le \|v_{n'} - s\|$. Here, n' is the number of incoming links at sink s, which transmits simultaneously to s; clearly, $n' \le n$. Next, we try to bound the SINR of the sink node s. For any link $v_i s$ ($i \ne 1$), its SINR

$$\text{SINR}_{is} \le \frac{P \cdot l(v_i, s)}{N_0 + \sum_{k=1}^{i-1} P \cdot l(v_k, s)} \le \frac{P \cdot l(v_i, s)}{N_0 + \sum_{k=1}^{i-1} P \cdot l(v_i, s)} < \frac{1}{i-1}.$$

■

Therefore, for $i \neq 1$,

$$W_{is} = W \log_2(1 + \text{SINR}_{is}) < W \log_2\left(\frac{i}{i-1}\right).$$

So the maximum rate at sink s is at most

$$W_{1s} + \sum_{i=2}^{n'} W \log_2\left(\frac{i}{i-1}\right) = W_{1s} + \log_2\left(\prod_{i=2}^{n'} \frac{i}{i-1}\right)$$

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

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 that is playing an important role depends on the relationship between n and $1 + \frac{P}{N_0}$. If P and N_0 are constants as we assumed, $\max_i(W_{is}) \le O(W)$. Then, the upper bound of capacity can be written as $O((\log n)\ W)$.

We now can introduce our data collection algorithm, which uses the same partition method and scheduling algorithm as that of the physical model. The only difference is the size of the interference block.

We now divide the field into big interference blocks of a certain size $L(d) \times L(d)$ as shown in Figure 1.7. Thus, the number of interference blocks is $\frac{l^2}{L(d)^2}$. In our collection scheme, we will schedule data transmission in parallel at all blocks but make sure that there is only one sensor in each interference block transferring at any time.

We now prove that the transmission rate of each transmitting sensor node in such data collection scheme is at least $\Omega\left((\log n)^{-\frac{\beta}{2}} W\right)$, if $L(d) = \kappa d$ and $\kappa > 2$ is a constant.

Lemma 5

In each interference block with size of $\kappa d \times \kappa d$ [25], there exists a node that can transmit at rate $\Omega\left((\log n)^{-\frac{\beta}{2}} W\right)$ to any destination in its adjacent cell. ■

Proof

Let us focus on one given sensor node v_i, which transmits to a destination v_j in v_i's adjacent cell. Its transmission rate is:

$$W_{ij} = W \log_2\left(1 + \frac{P \cdot l(v_i, v_j)}{N_0 + \sum_{k \in I} P \cdot l(v_k, v_j)}\right).$$

■

Because the distance between v_i and v_j is at most $\sqrt{5}d$, $P \cdot l(v_i, v_j) \geq P \cdot (\sqrt{5}d)^{-\beta} = \Omega(d^{-\beta})$.

We then need to find the upper bound of the interference at the receiver v_j from simultaneous transmitters. Using the same technique in Section 1.2.4.1, we consider layers of simultaneous transmissions in surrounding interference blocks as shown in Figure 1.7. Once again, assume that $d_i \geq iL - 2d$ is the minimum distance from an ith layer transmitter to v_j and $c_i = 8i$ is the number of transmitters on the ith layer. Therefore,

$$\sum_{k \in I} P \cdot l(v_k, v_j) \leq \sum_{i=1}^{\infty} 8iP[iL(d) - 2d]^{-\beta}$$

$$\leq \sum_{i=1}^{\infty} 8iP(i\kappa - 2)^{-\beta} d^{-\beta} \leq 8Pd^{-\beta} \cdot \sum_{i=1}^{\infty} i(i\kappa - 2)^{-\beta}.$$

Because $\beta > 2$, the summation $\sum_{i=1}^{\infty} i(i\kappa - 2)^{-\beta}$ converges to a constant ρ. Therefore,

$$\sum_{k \in I} P \cdot l(v_k, v_j) \leq 8P\rho \cdot d^{-\beta} = O(d^{-\beta}).$$

When $n \to \infty$, $d \to \infty$; hence, the SINR

$$\frac{P \cdot l(v_i, v_j)}{N_0 + \sum_{k \in I} P \cdot l(v_k, v_j)} = \Omega(d^{-\beta}).$$

Therefore, the transmission rate from v_i to v_j

$$W_{ij} = \Omega(d^{-\beta} W).$$

We use the same data collection scheme in Section 1.2.2. The total time we need to collect all the n packets is

$$T \leq \left[\left(\frac{\kappa d}{d}\right)^2 O(\log n)m + \left(\frac{\kappa d}{d}\right) O(\log n)m^2\right] \cdot \frac{b}{\Omega(Wd^{-\beta})}$$

$$\le O((\log n)m^2)\cdot\frac{b}{\Omega(Wd^{-\beta})}=\frac{O(n)}{\Omega(d^{-\beta})}\cdot t\le O\left(n(\log n)^{\frac{\beta}{2}}\right)\cdot t.$$

Thus, the achieved capacity of data collection under the generalized physical model is $\Omega\left((\log n)^{-\frac{\beta}{2}}W\right)$.

In summary, the bounds of data collection capacity can be summarized as the following:

Theorem 5

Under the generalized physical model [25], the capacity of data collection in random sensor networks is between $\Omega\left((\log n)^{-\frac{\beta}{2}}W\right)$ and $O((\log n)\ W)$. ■

1.3 Data Collection in Arbitrary Sensor Networks

We have studied the capacity of data collection on large-scale random WSNs. However, all results are based on a strong assumption that sensors are deployed randomly in an environment and the number of nodes n must be extremely large. Such an assumption is useful to simplify the analysis and derive nice theoretical limits, but may be invalid in many practical sensor applications. In most of the practical sensor applications, the sensor network is not uniformly deployed 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 section, we consider an arbitrary WSN in which n sensors and a single sink s are arbitrarily deployed in a finite geographical region. Figure 1.2 illustrates the difference between a random network and an arbitrary network.

1.3.1 Data Collection under the Protocol Model

Recall that the upper bound of data collection capacity in random networks is W. Obviously, this upper bound also holds for any arbitrary networks because sink s cannot receive at a rate faster than W due to the fixed transmission rate at each link. Therefore, we now introduce a simple breadth first search (BFS) tree-based data collection scheme to achieve capacity in the same order of the upper bound, that is, $\Theta(W)$. The data collection method includes two steps: data collection tree formation and data collection scheduling.

1.3.1.1 Data Collection Tree: BFS Tree

The data collection tree used in our method is a classic BFS tree rooted at the sink s. The time complexity to construct such a BFS tree is $O(|V| + |E|)$. Let T be the BFS tree and $v_1^l, \ldots, v_c^l$ be all leaves in T. For each leaf v_i^l, there is a path P_i from itself to the root s. Let $\delta^{P_i}(v_j)$ be the number of nodes on path P_i that are inside the interference range of v_j (including v_j itself). Assume the maximum interference number 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.

Lemma 6

Given a BFS tree T under the protocol model [26,27], the maximum interference number Δ_i on each path P_i is bounded by a constant $8\alpha^2$, that is, $\Delta_i \le 8\alpha^2$. ■

Proof

We prove by contradiction with a simple area argument. Assume that there is a v_j on P_i whose $\Delta_i > 8\alpha^2$. In other words, more than $8\alpha^2$ nodes on P_i are located in the interference region of v_j. Because the area of interference region is πR^2, we consider the number of interference nodes inside a small disk with a radius of $\frac{r}{2}$ (see Figure 1.8 for illustration). The number of such small disks is at most $\frac{\pi R^2}{\pi\left(\frac{r}{2}\right)^2} = 4\alpha^2$ 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_x, v_y, and v_z on the path P_i are connected to each other as shown in Figure 1.8. This is a contradiction with the construction of the BFS tree. As shown in Figure 1.8, if v_x and v_z are connected in G, then v_z should be visited by v_x not v_y during the construction of the BFS tree. This finishes our proof. ■

1.3.1.2 Branch 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.

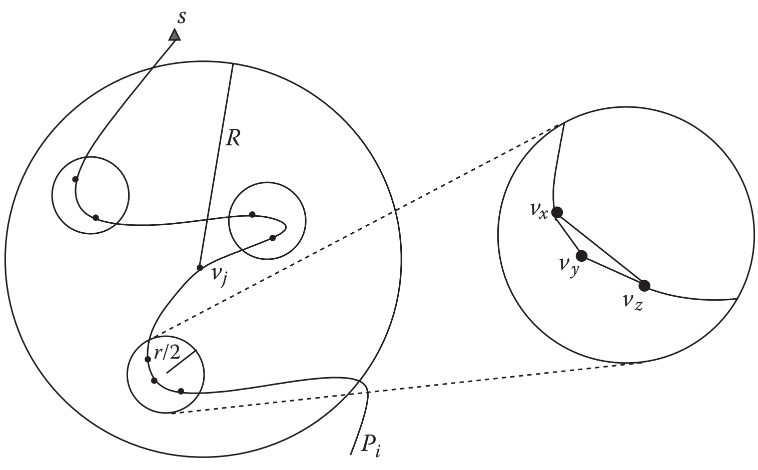

Figure 1.8 Proof of Lemma 6: on a path P_i in BFS tree T, the interference nodes for a node v_j is bounded by a constant.

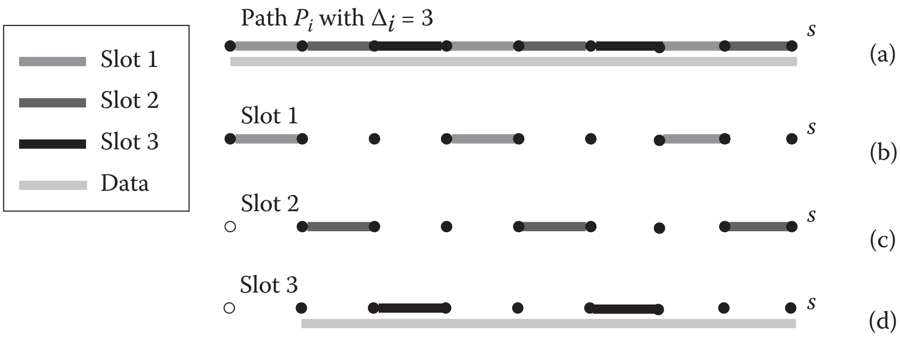

Figure 1.9 Scheduling on a path: after Δ_i slots, the sink obtains one set of data.

First, we explain how to schedule the collection on a single path. For a given path P_i, we can use Δ_i slots to collect one unit of data in the snapshot at the sink (see Figure 1.9 for illustration). In this figure, we assume that $R = r$, that is, only adjacent nodes interfere with each other. Thus, $\Delta_i = 3$. Then, we color the path using three shades as in Figure 1.9a. Notice that each node on the path has unit data to transfer. Links with the same color are active in the same slot. After three slots (Figure 1.9d), the leaf node has no data in this snapshot and the sink received one unit of 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 that T has c leaves, which define c paths from P_1 to P_c. Our algorithm collects data from path P_1 to P_c in order. We define the ith branch B_i as part of P_i from v_i^l to the intersection node with P_{i+1} for $i = (1, c - 1)$ and cth branch $B_c = P_c$. For example, in Figure 1.10b, there are four branches in T: B_1 is from v_1^l to v_a, B_2 is from v_2^l to s, B_3 is from v_3^l to v_b, and B_4 is from v_4^l to s. Notice that the union of all branches is the whole tree T. Algorithm 1 (in Figure 1.11) shows the detailed branch scheduling algorithm. Figure 1.10c through j gives an example of scheduling on T. In the first step (Figure 1.10c), all nodes on P_1 participate in the collection using the scheduling method for a single path (for every Δ_1 slot, sink s receives one unit of data). Such a collection stops until there is no data in this snapshot on branch B_1 (as shown in Figure 1.10d). Then step 2 collects data on path P_2. This procedure is repeated until all data in this snapshot reaches s (Figure 1.10j).

1.3.1.3 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 6

The data collection method based on path-scheduling in the BFS tree can achieve a data collection capacity of $\Theta(W)$ at the sink [26,27].

Proof

In Algorithm 1, the sink collects data from all c paths in T. In each step (lines 3 and 4), data are transferred on path P_i and it takes at most $\Delta_i \times |B_i|$ time slots. Recall that path scheduling

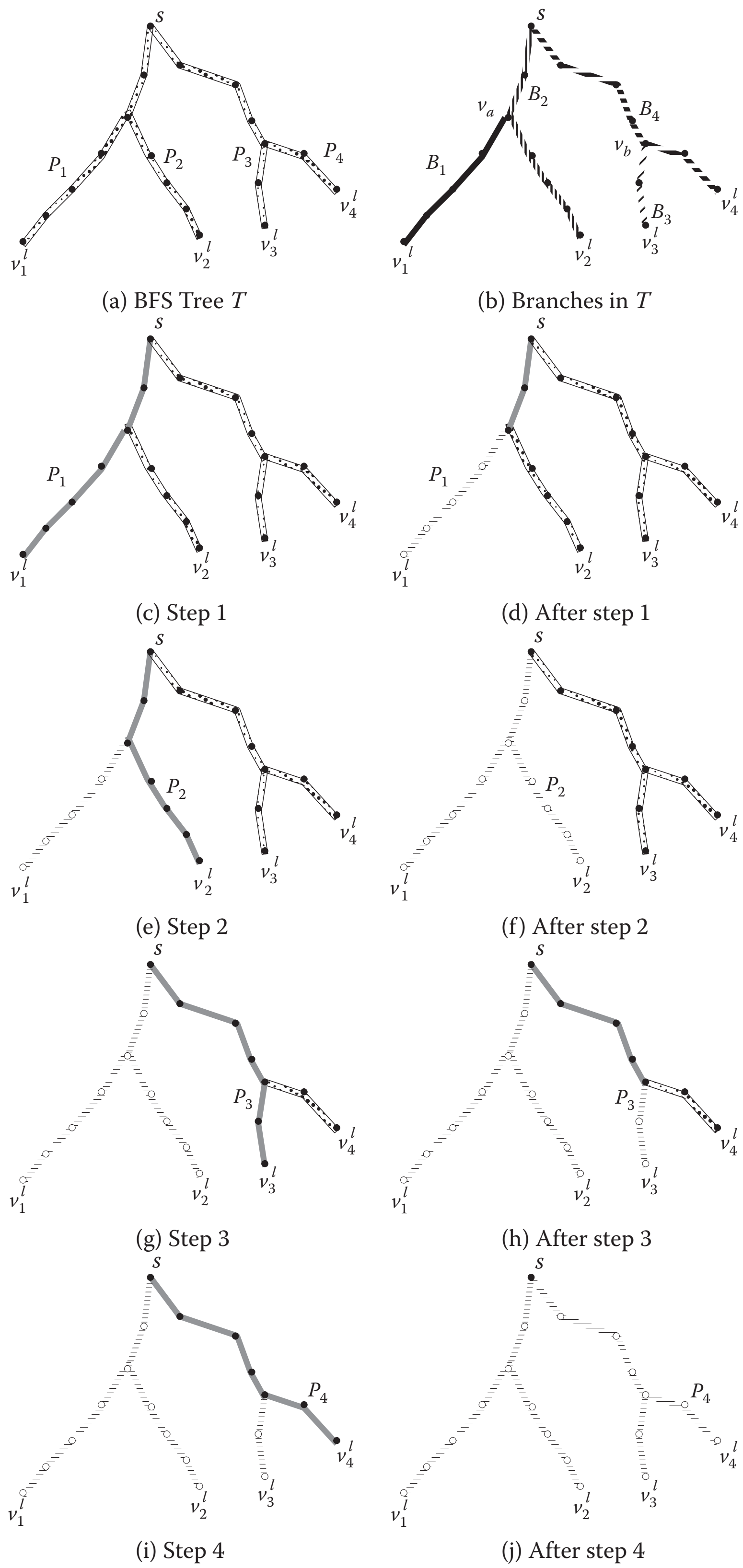

Figure 1.10 Illustrations of our scheduling on the data collection tree *T*.

Algorithm 1 Branch Scheduling on BFS Tree

Input: BFS tree T rooted at s.

for each snapshot **do**
    for $t = 1$ to c **do**
        Collect data on path P_i. All nodes on P_i transmit data towards the sink s using *path scheduling*.
        The collection terminates when nodes on branch B_i do not have data for this snapshot. The total slots used are at most $\Delta_i \cdot |B_i|$, $|B_i|$ is the hop length of B_i.
    end for
end for

Figure 1.11 Branch-scheduling algorithm on a BFS tree.

needs at most $\Delta_i \times k$ 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 $\sum_{i=1}^{c}\left(\Delta_i \times |B_i|\right)$. Because the union of all branches is the whole tree T, that is, $\sum_{i=1}^{c}|B_i| = n$, $\tau \leq \sum_{i=1}^{c}\left(\Delta_i \times |B_i|\right) \leq \sum_{i=1}^{c}\left(\tilde{\Delta} \times |B_i|\right) = \tilde{\Delta} n$. Here, $\tilde{\Delta} = \max\{\Delta_1, \cdots, \Delta_c\}$. Then, the delay of data collection $D = \tau t \leq \tilde{\Delta} nt$. The capacity $C = \frac{nb}{D} \geq \frac{nb}{\tilde{\Delta} nt} \leq \frac{W}{\tilde{\Delta}}$. From Lemma 6, we know that $\tilde{\Delta}$ is bounded by a constant. Therefore, the data collection capacity is $\Theta(W)$.

Remember 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 7

Under the protocol and disk graph models [26,27], data collection capacity for arbitrary WSNs is $\Theta(W)$. ■

1.3.2 Data Collection under the General Graph Model

In previous parts of this chapter, our collecting algorithm and analysis was based on a disk graph model in which two nodes could communicate if and only if their distance was less than or equal to the transmission range r. However, a disk graph model is idealistic because, in practice, two nearby nodes may be unable to communicate due to various reasons such as barriers and path fading. Therefore, in this subsection, we consider a more general graph model $G = (V, E)$ in which 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 correctly receive the signal from v_i is $\|v_i - v_j\| \leq r$. However, $\|v_i - v_j\| \leq r$ 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 be any general graph (for example, setting $r = \infty$ and putting a barrier between any two nodes v_i and v_j if $v_iv_j \notin G$).

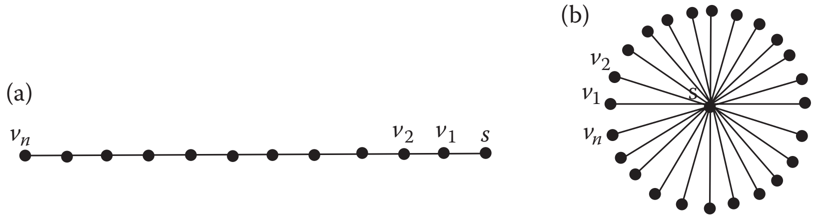

Figure 1.12 Two extreme cases in general graph models: (a) straight-line topology and (b) star topology.

In the 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 1.12a. Assume that the sink s is located at the end of the network and the interference range is large enough to cover every node in the network. Because the transmission on one link will interfere with all the other nodes, the only possible scheduling is transferring data along the straight-line via all links. The total time slots needed are $\frac{n(n+1)}{2}$, thus the capacity is at most $\frac{nb}{\frac{n(n+1)}{2}t} = \Theta\left(\frac{W}{n}\right)$. Notice that in this example, the maximum interference number 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 s in the center, as shown in Figure 1.12b. Clearly, a scheduling that lets every node transfer data in order can lead to a capacity W, which is much larger than $\frac{W}{\Delta} = \frac{W}{n}$. From these two examples, we find that the capacity problem for the general graph model is more complex. Next, we analyze the upper and lower bounds of the collection capacity under the protocol model for the general graph model.

1.3.2.1 Upper Bound of Collection Capacity

We first present a tighter upper bound of data collection capacity for the general graph model than the natural one W. Consider all packets from one snapshot, we use p_i to represent the packet generated by sensor v_i. For any v_i, let $l(v_i)$ be its level in the BFS tree rooted at the sink s (which is the minimum number of hops required for packet p_i or a packet at v_i to reach s). We use $D(s,l)$ to represent a virtual disk centered at the sink node s with a radius of hop distance l. The *critical level* (or the *critical radius*) l^* is the greatest level l such that no two nodes within l level from the sink node s can receive a message in the same time slot, that is, $l^* = \max\{l|\forall v_i, v_j \in D(s,l)$ cannot receive packets at the same time$\}$. The region defined by $D(s,l^*)$ is called the *critical region* (see Figure 1.13 for illustration). For any packet p_i originating at node v_i, we define

$$\lambda_i^* = \begin{cases} l(v_i), & \text{if } v_i \in < D(s,l^*), \\ l^*+1, & \text{otherwise.} \end{cases}$$

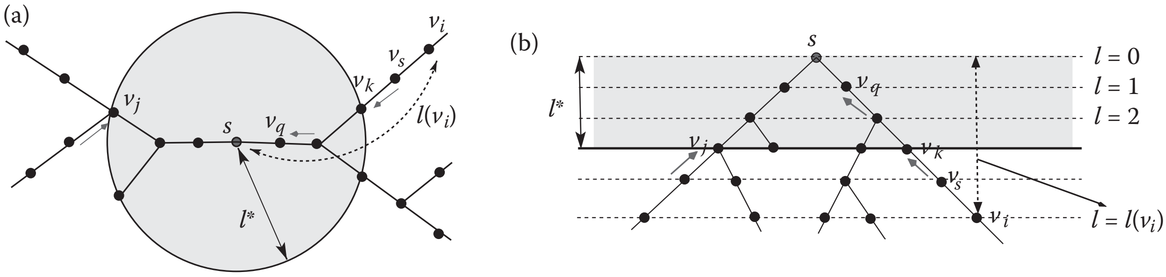

Figure 1.13 Illustration of the definition of the critical region, that is, *l. The gray area is the critical region, where no two nodes can receive a message in the same time slot due to interference around s. Critical region around sink s (a) and a tree view of the critical region (b).**

Here, λ_i^* gives the minimum number of hops needed to reach the sink s after packet p_i reaches the critical region around s. Let $\lambda^* = \max_i\{\lambda_i^*\}$. Then, we can prove the following lemma on the lower bound of delay for data collection.

Lemma 7

For all packets from one snapshot [27], the delay to collect them at sink s

$$D \geq t\sum_i \lambda_i^*.$$

■

Proof

It is clear that the critical region around sink s is a bottleneck for the delay. Any packet inside the critical region can only move one step at each time slot. First, the total delay must be larger than the delay, which is needed for the case in which all packets originating from outside the critical region are just one hop away from the critical region. In other words, assume that we can move all packets originating from outside the critical region to the surrounding area without spending any time. Then, each packet p_i needs λ_i^* time slots to reach the sink. By the definition of the critical region, no simultaneous transmissions around the critical region (one hop away from it) can be scheduled in the same slot. Therefore, the delay is at least the summation of λ_i^*. ■

Let $\Delta^* = \frac{\sum_i \lambda_i^*}{n}$, we have a new upper bound of data collection capacity, $C \leq \frac{W}{\Delta^*} \leq W$. Notice that $\Delta^* \geq 1$ and it represents the limit of scheduling due to interference around the sink (and its critical region).