PART 1

Sensor Networks

COPYRIGHTED MATERIAL

1

Fluid Models and Energy Issues

Wireless sensor networks consist of hundreds to thousands of sensor nodes with limited computational and energy resources. Sensors are densely deployed over an area of interest, where they gather and disseminate local data using multi-hop communications, i.e. using other nodes as relays. A typical network configuration includes a large collection of stationary sensors operating in an unattended mode, which need to send their data to a node which collects the networks' information, the so-called *sink node*.

Traditionally, network designers have used either computer simulations or analytical frameworks to predict and analyze a system's behavior. Modeling large sensor networks, however, raises several challenges due to scalability problems and high computational costs. With regard to simulations, several software tools have been extended and developed to deal with large wireless networks, see [ZEN 98, SIM 03, LEV 03] just to name a few. As for analytical modeling, to the best of our knowledge, the only work dealing with large sensor networks is presented in [DOU 04], which employs percolation techniques.

This chapter presents spatial fluid-based models for the analysis of large-scale wireless networks. The technique is said to be *fluid-based* because it represents the sensor nodes as a fluid entity. Sensor location is smoothed out in continuous space by introducing the concept of *local sensor density*, i.e. the number of sensors per area unit at a given point.

The approach is applied to describe a network scenario where nodes are static and need to send the result of their sensing activity to a sink node.

Sensors may send packets to the sink in a multi-hop fashion. Although this technique requires the introduction of simplified assumptions, that are necessary to maintain the problem tractable, these models account for (1) node energy consumption, (2) node contention over the radio channel and (3) traffic routing.

By the end of the chapter, three fundamental contributions are provided with respect to existing literature:

1) because of the fluid approach, very large networks can be studied while maintaining the model complexity extremely low;

2) the behavior of the network can be studied as a function of the bidimensional spatial distribution of the nodes, possibly under non-homogeneous node deployment;

3) the approach provides a very flexible and powerful tool, which can account for various routing strategies, sensor behaviors and network control schemes, such as congestion control mechanisms.

1.1. The fluid-based approach

The fluid approach is motivated by the observation that large-scale sensor networks can be represented by a continuous fluid entity distributed on the network area. This section describes the general framework, and the notation used to specify the model is summarized in Table 1.1.

Notation	Description
$\rho(\boldsymbol{r})$	Sensor density at $\boldsymbol{r}$
$\lambda(\boldsymbol{r})$	Local traffic generation rate density at $\boldsymbol{r}$
$\Lambda(\boldsymbol{r})$	Total traffic rate density at $\boldsymbol{r}$
$\Lambda^{\star}(\boldsymbol{r})$	Actual total traffic rate density at $\boldsymbol{r}$
$u(\boldsymbol{r}'\|\boldsymbol{r})$	Probability density of routing a packet from $\boldsymbol{r}$ to $\boldsymbol{r}'$
$s(\boldsymbol{r})$	Mean packet service time at $\boldsymbol{r}$
$q(\boldsymbol{r})$	Mean queueing delay at $\boldsymbol{r}$
$D(\boldsymbol{r})$	Mean delivery delay at $\boldsymbol{r}$
$P_R(\boldsymbol{r})$	Mean packet retransmission probability at $\boldsymbol{r}$
P_a	Probability that a sensor is active

Table 1.1. *Model notation*

1.1.1. *Sensor density and traffic generation*

Sensors are randomly placed over an area in the plane according to a Poisson point process with local intensity $\rho(\boldsymbol{r})$, hereinafter also called the *sensor density*, which can vary from point to point. Let us identify each point in the plane by means of its coordinates $\boldsymbol{r} = (x, y)$.

The Poisson assumption implies that the number of sensors contained in an area A is distributed according to a Poisson distribution with parameter $\Gamma(A)$, defined as:

$$\Gamma(A) = \iint_A \rho(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}.$$

The mean number of sensors present in the network is denoted by N, with $\iint \rho(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r} = N$. As an example, to define a system where there are (an average of) N sensors uniformly distributed over a disk of unit radius and the sink is located at the center of the disk (i.e. $\mathbf{Sink} = (0, 0)$), it is correct to write:

$$\rho(\boldsymbol{r}) = \begin{cases} 0 & \text{if } dist(\boldsymbol{r}, \mathbf{Sink}) > 1 \\ \dfrac{N}{\pi} & \text{if } dist(\boldsymbol{r}, \mathbf{Sink}) \leq 1 \end{cases} \qquad [1.1]$$

Finally, it is fair to assume that a sensor s in position $\boldsymbol{r}$ generates traffic at rate $\lambda_s(\boldsymbol{r})$. By aggregating all traffic generated by sensors over an infinitesimal area centered at point $\boldsymbol{r}$, the generation rate density is defined as $\lambda(\boldsymbol{r})$, which depends on the position $\boldsymbol{r}$. This quantity, measured in packets per second per area unit, is proportional to both the local generation rate of a sensor and the local sensor density and corresponds to the mean number of packets per second generated by an infinitesimal area. It is defined as:

$$\lambda(\boldsymbol{r}) = \lambda_s(\boldsymbol{r})\rho(\boldsymbol{r}) \qquad [1.2]$$

1.1.2. *Data routing*

The next hop used by a sensor to send a packet to the sink is determined in a probabilistic way. Indeed, the exact location of the sensors is unknown, thus

$u(\boldsymbol{r'}|\boldsymbol{r})$ can be defined as the probability density that a packet transmitted by a sensor in position $\boldsymbol{r}$ uses a sensor in position $\boldsymbol{r'}$ as its next hop. Since $u(\boldsymbol{r'}|\boldsymbol{r})$ must be a valid probability density, it is correct to have:

$$\iint u(\boldsymbol{r'}|\boldsymbol{r})\,\mathrm{d}\boldsymbol{r'} = 1, \quad \forall \boldsymbol{r} \qquad [1.3]$$

Probability density $u(\boldsymbol{r'}|\boldsymbol{r})$ depends on the particular routing policy.

1.1.3. *Local and relay traffic rates*

Each sensor can be both a traffic source and a relay for other sensors. The *traffic rate density* $\Lambda(\boldsymbol{r})$ is equal to the sum of the traffic locally generated by the sensors at point $\boldsymbol{r}$, and the traffic relayed for other nodes. By assuming that the system is stable, the total traffic rate density $\Lambda(\boldsymbol{r})$ can be computed by solving the following integral equation:

$$\Lambda(\boldsymbol{r}) = \lambda(\boldsymbol{r}) + \iint \Lambda(\boldsymbol{r'})u(\boldsymbol{r'}|\boldsymbol{r})\,\mathrm{d}\boldsymbol{r'} \qquad [1.4]$$

where $\lambda(\boldsymbol{r})$ accounts for the traffic locally generated, and the integral computes the rate density of the relayed traffic using $u(\boldsymbol{r'}|\boldsymbol{r})$ introduced above. Note that the expression in [1.4] represents the traffic rate density of successfully transmitted packets. The *actual traffic rate density* must account also for retransmissions, as explained in the following.

1.1.4. *Channel contention and data transmission*

The channel contention model computes the actual traffic rate density $\Lambda^{\star}(\boldsymbol{r})$ at a node in $\boldsymbol{r}$, as well as the mean packet service time at the same point, denoted by $s(\boldsymbol{r})$. Packets that are not received correctly need to be retransmitted by the sender. The average packet retransmission probability at $\boldsymbol{r}$ is denoted by $P_R(\boldsymbol{r})$: it depends on the particular protocol adopted to access the channel and is, in general, location dependent, i.e. it can be different from point to point within the network area. By assuming that the

packet transmission process is memoryless, the actual traffic rate density at point $\boldsymbol{r}$, which also accounts for retransmitted packets, is given by

$$\Lambda\star(\boldsymbol{r}) = \frac{\Lambda(\boldsymbol{r})}{1 - P_R(\boldsymbol{r})} \qquad [1.5]$$

1.1.5. *Mean packet delivery delay*

To compute the mean time needed to deliver a packet to the sink, the *mean delivery delay* is introduced at point $\boldsymbol{r}$, $D(\boldsymbol{r})$, and is defined as the time required by a packet originated in $\boldsymbol{r}$ to reach the sink. By denoting with $q(\boldsymbol{r})$ the mean queueing delay experienced by a packet at point $\boldsymbol{r}$, $D(\boldsymbol{r})$ can be expressed as

$$D(\boldsymbol{r}) \quad = \quad q(\boldsymbol{r}) + s(\boldsymbol{r}) + \iint D(\boldsymbol{r'})u(\boldsymbol{r'}|\boldsymbol{r})\,\mathrm{d}\boldsymbol{r'} \qquad [1.6]$$

where $s(\boldsymbol{r})$ is the mean service time previously introduced. Equation [1.6] states that the mean delivery delay at point $\boldsymbol{r}$ can be expressed as the sum of the delay experienced by a packet at point $\boldsymbol{r}$ plus the mean delivery delay associated with the next hop. The delivery delay in all different points of the network can be computed recursively starting from $D(0,0) = 0$, i.e. no delay is experienced by a packet at the sink.

1.1.6. *Sensor active/sleep behavior*

The fluid model accounts for the active/sleep dynamics of the nodes by introducing the probability P_a that a sensor is active. Since only active sensors generate traffic, [1.2] becomes $\lambda(\boldsymbol{r}) = \lambda_s(\boldsymbol{r})P_a\rho(\boldsymbol{r})$, where the sensor density $\rho(\boldsymbol{r})$ has been multiplied by the probability that a sensor is active.

1.2. Network scenario

The fluid-based modeling is applied to the sensor characteristics and the network scenario presented in this section together with the assumptions made to describe the system under study. The system parameters are summarized in Table 1.2.

Parameter	Description
N	Mean number of sensors
d_c, d_s	Maximum communication/carrier sensing ranges
P_i	Power consumed by idle sensors
$E^{(tx)}, E^{(rx)}$	Energy spent to transmit/receive one packet
$\delta_s, \delta_a, \sigma$	Time constants for the CSMA/CA scheme
CW	Contention window for the CSMA/CA scheme
K_{Max}	Maximum number of next hops for minimum energy routing

Table 1.2. *System parameters*

1) *Communication range*: All nodes have a common, maximum communication range equal to d_c. Thus, any pair of nodes, say (i, j), can communicate if they are within distance d_c from each other, i.e. $\mathrm{dist}(i, j) \leq d_c$, where the notation $\mathrm{dist}(i, j)$ denotes the Euclidean distance between nodes i and j. It is assumed that, when a sensor discovers a neighboring node and a wireless link is established between the two nodes (i.e. a connection at the link layer is created), the sensors can set their level of transmission power to be used over that link. Note that sensors do not adjust over time their transmission energy toward a neighbor, but rather the transmission level used by a node over a link is set once upon the link establishment and remains unchanged with time. This is a fair assumption since typically sensor nodes are simple, low-cost devices.

2) *Error model*: The communication channel is assumed to be error-free, although a channel error process could be easily included in the model.

3) *Topology*: The network topologies are considered to be always connected, i.e. there exists at least one path connecting each sensor to the sink. Moreover, it is assumed that nodes that cannot reach the sink do not participate in the network operation, thus they can be simply neglected.

4) *Data generation*: Each sensor generates data packets of constant size at a given rate, which can be buffered while waiting for transmission. The system is considered stable and the buffer of each sensor is modeled as a first-in first-out queue. Also, by considering that the buffer is properly dimensioned so that the loss probability due to overflow is negligible, it is fair to assume infinite buffer capacity, though the case of finite buffer size could be easily incorporated in the model.

5) *Power consumption*: Sensors consume a power equal to P_i while idle; the energy expenditure due to a one-hop communication is modeled as follows.

Given the transmitter–receiver pair (i, j), the energy consumed by i to transmit a packet to j is equal to $E^{(tx)}(i, j)$ and the energy consumed by j to receive a packet is equal to $E^{(rx)}$; the energy consumed while overhearing[1] is assumed to be equal to $E^{(rx)}$. Note that a sensor cannot simultaneously transmit and receive, and the energy expenditures while transmitting/receiving/overhearing are additional with respect to the energy spent by a sensor while idle.

6) *Channel access scheme*: Sensors access the channel by using a carrier sense multiple access scheme with collision avoidance (CSMA/CA). The carrier sensing range of each node is denoted by d_s, where $d_s \geq d_c$, i.e. the carrier sensing range is larger than or equal to the communication range. When a node has to transmit a packet, it senses the medium for a time interval δ_s. If idle, the sensor accesses the channel. If busy, the sensor waits for the channel to become idle; then, within an interval δ_s, the node selects a backoff counter uniformly within its current contention window (CW). The backoff counter is decremented by one after the channel has been idle for the duration of a slot σ, whereas it is frozen while the channel is busy. When the backoff counter reaches zero, the sensor attempts to transmit. If a collision occurs, the sensor doubles CW and repeats the procedure. If the data transmission is successful, the sensor receives an acknowledgment message (ACK) from the receiver after a time interval equal to δ_a (with δ_a shorter than δ_s), resets CW to the minimum CW and extracts a new backoff value (the so-called post backoff). Note that no handshaking messages are employed (i.e. request to send/clear to send are not used). A node attempts to transmit a packet until the packet transmission is successful; considering that buffers are of infinite capacity, this assumption implies that all packets eventually reach the sink. Power capture effects are not considered: whenever a node receives two or more packets sent by nodes within its carrier sensing range, all of them are lost. On the other hand, if a node senses a single packet, it can always receive it successfully, since we assume an error-free channel (assumption (2)). Indeed, in CSMA networks the MAC protocol ensures that, once a node gains access to the channel, there is no interference in the absence of collisions. Finally, for analytical tractability multiple collisions are neglected, i.e. if a packet transmission fails because of a collision, the next attempt at transmitting that packet will be successful (in

1 A node is in *overhearing* mode when it listens to a packet transmitted over the channel for which it is not the intended destination.

the network scenario under study this is a reasonable assumption, as shown in section 1.3.2).

7) *Sensor sleep mode*: Sensors may enter a low power operational state (sleep mode), while in the sleep state it is assumed that their sensor board and radio frequency circuitry are turned off, i.e. they do not generate, receive or transmit any traffic. The sleep/activity cycles of the sensors are asynchronous, and, when nodes enter or exit the sleep state, they announce the change in operational mode to their neighbors. Note that, although sleep/activity cycles for the sensors are asynchronous, it is supposed that at each time instant there exists at least one active node among the sensors covering a certain area. This is a fair assumption due to the large number of nodes composing the network. Asynchronous schemes therefore give good performance, without adding complexity to the control plane as maintaining synchronization among sensors would require.

8) *Multi-hop communication*: To deliver their data, sensors may use multi-hop paths, specifically minimum energy routing. When sensors are used on the field, they undergo an initial configuration phase in which they compute once and for all the next best available hops to reach the sink, using a variation of the standard Bellman–Ford algorithm [BER 92]. In particular, each sensor maintains an ordered routing table of up to K_{Max} entries, each associated with a different next hop, which record the total cost required to reach the sink by following the corresponding next hop. The tables can be constructed in a distributed fashion using a controlled flooding algorithm: starting from the sink, sensors broadcast their routing tables to the neighbors, which in turn update their tables and, if there is a change in at least one entry, rebroadcast them. After a few iterations, the mechanism converges (i.e. the routing table at each sensor does not change anymore). In the case of minimum energy routing, an energy cost is assigned to each link connecting two nodes that are within distance d_c from each other. The cost $\epsilon(i,j)$ represents the total energy required to transfer a packet from node i to node j (one-hop energy cost) and is expressed by the sum of the cost at the transmitter and the cost at the receiver: $\epsilon(i,j) = E^{(tx)}(i,j) + E^{(rx)}$. Due to assumption (7), the next hop node that minimizes the total energy cost to reach the sink can be in a low operational state, and thus not be available. In this case, we assume that the transmitting node will use the second best minimum energy next hop. If this is not active either, the routing algorithm will select the third one, and so on, up to a maximum number of alternative next hops K_{Max}. We assume that K_{Max}

is sufficiently large so that the probability that all K_{Max} next hop sensors are in the low-power operational state is negligible.

1.3. The sensor network model

The framework outlined in section 1.1 requires several functions to be defined to account for the particular characteristics of the network scenario considered in the previous section. In particular:

– section 1.3.1 specifies $u(\boldsymbol{r'}|\boldsymbol{r})$ for the case of minimum energy shortest path routing;

– to account for channel contention using the CSMA/CA protocol, section 1.3.2 presents a detailed expression for the mean packet service time $s(\boldsymbol{r})$ and packet retransmission probability $P_R(\boldsymbol{r})$ at a node in $\boldsymbol{r}$;

– section 1.3.3 introduces a queueing model to estimate the mean queueing delay experienced by a packet at $\boldsymbol{r}$ (i.e. $q(\boldsymbol{r})$), which accounts for the sensors' active/sleep dynamics. Recall that $q(\boldsymbol{r})$ is required to compute the mean time needed to deliver a packet to the sink starting at point $\boldsymbol{r}$ (i.e. $D(\boldsymbol{r})$).

1.3.1. *A minimum energy routing strategy: computing* $u(\boldsymbol{r'}|\boldsymbol{r})$

To model the minimum energy routing scheme defined in section 3.1, it is necessary to specify the routing function $u(\boldsymbol{r'}|\boldsymbol{r})$, taking into account the fact that if the minimum energy next hop is in the low power operational state, then the second best next hop is selected, and so on. The definition of $u(\boldsymbol{r'}|\boldsymbol{r})$ is carried out through successive steps where auxiliary functions are defined. In particular, the following derivations are computed:

1) the *multi-hop energy cost*, denoted by $\epsilon_m(\boldsymbol{r}, \boldsymbol{r'})$, which is the energy required to send a packet from point $\boldsymbol{r}$ to the sink using position $\boldsymbol{r'}$ as next hop. Note that this cost is defined regardless of the presence of sensors in positions $\boldsymbol{r}$ and $\boldsymbol{r'}$;

2) the cumulative probability $F^k_{mE}(e|\boldsymbol{r})$ that the energy required to send a packet from a sensor in $\boldsymbol{r}$ to the sink using the k-th lowest energy route is less than or equal to e;

3) the cumulative probability $F_{mE}(e|\boldsymbol{r})$ that the minimum energy required to send a packet from a sensor in $\boldsymbol{r}$ to the sink, using only active next hops, is less than or equal to e;

4) the probability density $p_{s;\mathbf{r}}(\boldsymbol{r'}|e)$ of finding a sensor in position $\boldsymbol{r'}$ that can be used as next hop by a sensor in $\boldsymbol{r}$ to send a packet to the sink with energy expenditure e, conditioned to the fact that e is the minimum required energy (using only active nodes);

5) the probability density $u(\boldsymbol{r'}|\boldsymbol{r})$ that a packet generated by a sensor in position $\boldsymbol{r}$ uses a sensor in position $\boldsymbol{r'}$ as its next hop.

1.3.1.1. *Computing* $\epsilon_m(\boldsymbol{r}, \boldsymbol{r'})$

The *one-hop energy cost* $\epsilon(\boldsymbol{r}, \boldsymbol{r'})$ required to deliver a packet from a source in $\boldsymbol{r}$ to a destination in $\boldsymbol{r'}$ is defined as

$$\begin{aligned} \epsilon(\boldsymbol{r}, \boldsymbol{r'}) &= E^{(tx)}(\boldsymbol{r}, \boldsymbol{r'}) + E^{(rx)} \\ &= 2\left(E^{(\text{ele})} + E^{(\text{proc})}\right) + C_d \cdot \text{dist}(\boldsymbol{r}, \boldsymbol{r'})^{\eta} \end{aligned} \quad [1.7]$$

where $E^{(\text{ele})}$ and $E^{(\text{proc})}$ account for the consumption due to the transceiver electronics and to processing functions, respectively. These costs are present at both the transmitter and the receiver; the amplifier cost affects only the transmitter, and includes a constant factor C_d, the sender–receiver distance $\text{dist}(\boldsymbol{r}, \boldsymbol{r'})$ and the exponential power decay factor η, that typically takes values between 2 and 4 [RAP 96].

By using a recursive expression, it can be defined:

$$\epsilon_m(\boldsymbol{r}, \boldsymbol{r'}) = \begin{cases} \min\left(\epsilon(\boldsymbol{r}, \mathbf{Sink}), \epsilon(\boldsymbol{r}, \boldsymbol{r'}) + \epsilon_m(\boldsymbol{r'}, \boldsymbol{r''})\right) & \text{if } dist(\boldsymbol{r}, \mathbf{Sink}) \leq d_c \\ \epsilon(\boldsymbol{r}, \boldsymbol{r'}) + \epsilon_m(\boldsymbol{r'}, \boldsymbol{r''}) & \text{otherwise} \end{cases} \quad [1.8]$$

where $\boldsymbol{r''}$ is the point that minimizes the energy required to send a packet from $\boldsymbol{r'}$ to the sink. Note that, if point $\boldsymbol{r}$ is within distance d_c from the sink, either a one-hop or a multi-hop communication may take place, depending on their energy cost. When a multi-hop communication is required, the definition of $\boldsymbol{r''}$ turns out to be exceedingly complicated. Thus, $\epsilon_m(\boldsymbol{r'}, \boldsymbol{r''})$ is aproximated with $\epsilon_{\min}(\boldsymbol{r'})$ that is the minimum possible energy needed to send a packet from $\boldsymbol{r'}$

to the sink. As described in the following section, $\epsilon_{\min}(\boldsymbol{r'})$ can be computed as:

$$\epsilon_{\min}(\boldsymbol{r'}) = 2h\left(E^{(\text{ele})} + E^{(\text{proc})}\right) + C_d \frac{\text{dist}(\boldsymbol{r'}, \textbf{Sink})^{\eta}}{h^{\eta-1}} \quad [1.9]$$

with

$$h = \max\left(h^*, \left\lceil \frac{\text{dist}(\boldsymbol{r'}, \textbf{Sink})}{d_c} \right\rceil\right)$$

where

$$h^* = \arg\min_{k\in\mathbb{N}} \left[2k\left(E^{(\text{ele})} + E^{(\text{proc})}\right) + C_d \frac{\text{dist}(\boldsymbol{r'}, \textbf{Sink})^{\eta}}{(h^*)^{\eta-1}}\right]$$

By inserting [1.9] into [1.8], $\epsilon_m(\boldsymbol{r}, \boldsymbol{r'})$ can be rewritten as

$$\epsilon_m(\boldsymbol{r}, \boldsymbol{r'}) = \begin{cases} \min\left(\epsilon(\boldsymbol{r}, \textbf{Sink}), \epsilon(\boldsymbol{r}, \boldsymbol{r'}) + \epsilon_{\min}(\boldsymbol{r'})\right) & \text{if } \text{dist}(\boldsymbol{r}, \textbf{Sink}) \leq d_c \\ \epsilon(\boldsymbol{r}, \boldsymbol{r'}) + \epsilon_{min}(\boldsymbol{r'}) & \text{otherwise} \end{cases} \quad [1.10]$$

Equation [1.10] states that, in case of multi-hop communication, the energy required to deliver a packet from $\boldsymbol{r}$ to the sink using $\boldsymbol{r'}$ as next hop is equal to the energy required to transfer a packet from $\boldsymbol{r}$ to $\boldsymbol{r'}$, plus the minimum energy required to send a packet from $\boldsymbol{r'}$ to the sink. Note that [1.10] is defined regardless of the presence of sensors in positions $\boldsymbol{r}$ and $\boldsymbol{r'}$.

1.3.1.2. *Computing* $F^k_{mE}(e|\boldsymbol{r})$

Given a sensor at point $\boldsymbol{r}$, the cumulative probability $F^k_{mE}(e|\boldsymbol{r})$ that the energy required to send a packet from the sensor to the sink, using a node that provides the kth minimum energy route, is less than or equal to e, is given by[2]:

$$F^k_{mE}(e|\boldsymbol{r}) \quad = \quad P\left\{\text{Poisson}\left(\iint_{\mathbf{r'}:\epsilon_m(\mathbf{r},\mathbf{r'})\leq e} \rho(\boldsymbol{r'})\, \mathrm{d}\boldsymbol{r'}\right) \geq k\right\} \quad [1.11]$$

2 *Poisson*(ν) represents a Poisson random variable with mean ν.

Equation [1.11] states that the probability of having the kth minimum energy path requiring expenditure less than or equal to e corresponds to the probability of finding at least k relay sensors in an area through which a packet can be transferred to the sink with energy cost less than or equal to e.

1.3.1.3. *Computation of the minimum energy path (equation [1.9])*

Here, it is proved that the minimum possible energy required to send a packet to the sink from a point $\boldsymbol{r}$ is given by [1.9]. The proof is carried on into three steps: first it is proved that all sensors on the route must be aligned, and then that the distance between two consecutive sensors must be the same; finally, it is shown how to determine the minimum number of hops required to reach the sink.

As a first step, assume that the minimum number of hops required to reach the sink from point $\boldsymbol{r}$ is known and equal to h. Let $\xi(\boldsymbol{r}) = (\boldsymbol{r}_0, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_k, \ldots \boldsymbol{r}_h)$ (with $\boldsymbol{r}_0 = \boldsymbol{r}$ and $\boldsymbol{r}_h = \mathbf{Sink}$) be a sequence of points, such that $\mathrm{dist}(\boldsymbol{r}_j, \boldsymbol{r}_{j+1}) \leq d_c$, $0 \leq j < h$. The energy required to send a packet from $\boldsymbol{r}$ to the sink using as relays the sensors located at the points of $\xi(\boldsymbol{r})$ is given by

$$\begin{aligned} E(\xi(\boldsymbol{r})) &= \sum_{j=0}^{h-1} \left[2\left(E^{(\mathrm{ele})} + E^{(\mathrm{proc})}\right) + C_d \cdot \mathrm{dist}(\boldsymbol{r}_j, \boldsymbol{r}_{j+1})^{\eta} \right] \\ &= 2h\left(E^{(\mathrm{ele})} + E^{(\mathrm{proc})}\right) + C_d \sum_{j=0}^{h-1} \mathrm{dist}(\boldsymbol{r}_j, \boldsymbol{r}_{j+1})^{\eta} \end{aligned}$$

It can be proved that on the minimum energy path all points in $\xi(\boldsymbol{r})$ stay on the line connecting $\boldsymbol{r}$ to the sink. Suppose for simplicity that all points are aligned except point $\boldsymbol{r}_k$. Let us call $\boldsymbol{r'}_k$ the projection of point $\boldsymbol{r}_k$ on the line. Notice that $\boldsymbol{r'}_k$ is aligned with the other points. Let us call $\xi'(\boldsymbol{r}) = (\boldsymbol{r}_0, \ldots, \boldsymbol{r'}_k, \ldots, \boldsymbol{r}_h)$. By construction, it is fair to write: $\mathrm{dist}(\boldsymbol{r}_{k-1}, \boldsymbol{r'}_k)^{\eta} \leq \mathrm{dist}(\boldsymbol{r}_{k-1}, \boldsymbol{r}_k)^{\eta}$ and $\mathrm{dist}(\boldsymbol{r'}_k, \boldsymbol{r}_{k+1})^{\eta} \leq \mathrm{dist}(\boldsymbol{r}_k, \boldsymbol{r}_{k+1})^{\eta}$. Then, it can be shown that

$$E(\xi'(\boldsymbol{r})) \leq E(\xi(\boldsymbol{r}))$$

$$E(\xi'(\boldsymbol{r})) = 2h\left(E^{(\mathrm{ele})} + E^{(\mathrm{proc})}\right) + C_d \cdot \left(\sum_{j=0, j\neq k-1,k}^{h-1} \mathrm{dist}(\boldsymbol{r}_j, \boldsymbol{r}_{j+1})^\eta + \mathrm{dist}(\boldsymbol{r}_{k-1}, \boldsymbol{r'}_k)^\eta + \mathrm{dist}(\boldsymbol{r'}_k, \boldsymbol{r}_{k+1})^\eta\right) \leq 2h\left(E^{(ele)} + E^{(\mathrm{proc})}\right) + C_d \cdot \left(\sum_{j=0, j\neq k-1,k}^{h-1} \mathrm{dist}(\boldsymbol{r}_j, \boldsymbol{r}_{j+1})^\eta + \mathrm{dist}(\boldsymbol{r}_{k-1}, \boldsymbol{r}_k)^\eta + \mathrm{dist}(\boldsymbol{r}_k, \boldsymbol{r}_{k+1})^\eta\right) = E(\xi(\boldsymbol{r}))$$

Now it can be assumed that all points of the route are aligned and it can be proved that the distance between two consecutive points is constant and equal to $\alpha = \frac{\mathrm{dist}(\boldsymbol{r}_0, \boldsymbol{r}_h)}{h}$. Consider a generic sequence $\xi(\boldsymbol{r})$ of h points aligned on the same segment. The distance between two consecutive points is written as: $\mathrm{dist}(\boldsymbol{r}_j, \boldsymbol{r}_{j+1}) = \alpha_j$, with $0 \leq j \leq h-2$, and the one between the last two points as: $\mathrm{dist}(\boldsymbol{r}_{h-1}, \boldsymbol{r}_h) = \mathrm{dist}(\boldsymbol{r}_0, \boldsymbol{r}_h) - \sum_{j=0}^{h-2} \alpha_j$. Then, it follows that

$$E(\xi(r)) = 2h(E^{ele} + E^{proc}) + C_d \left[\sum_{j=0}^{h-2} \alpha_j^\eta + \left(\mathrm{dist}(\boldsymbol{r}_0, \boldsymbol{r}_n) - \sum_{j=0}^{h-2} \alpha_j\right)^\eta\right]$$

The partial derivatives of $E(\xi(r))$ along α_j, $0 \leq j \leq h-2$, are as follows,

$$\frac{\partial E(\xi(r))}{\partial \alpha_j} = C_d \left[\eta \alpha_j^{\eta-1} - \eta\left(\mathrm{dist}(\boldsymbol{r}_0, \boldsymbol{r}_h) - \sum_{j=0}^{h-2} \alpha_j\right)^{\eta-1}\right]$$

It can be easily seen that all partial derivatives are equal to 0 if the distance between any two consecutive nodes α_j is constant and equal to $\alpha = \frac{\mathrm{dist}(\boldsymbol{r}_0, \boldsymbol{r}_h)}{h}$.

So far it has been proven that along the minimum energy path, all points are aligned and equidistant. If the number of points is equal to h, then it is fair to write

$$E_{\min}(\boldsymbol{r}, h)=2h\left(E^{(\text{ele})}+E^{(\text{proc})}\right)+C_d\frac{\text{dist}(\boldsymbol{r}, \mathbf{Sink})^{\eta}}{h^{\eta-1}} \qquad [1.12]$$

Now a valid value for h that minimizes the energy cost of the path has to be derived. If [1.12] is considered to be continuous on h, its derivative can be computed and it is easy to obtain the minimum of the function for $h = \text{dist}(\boldsymbol{r}_0, \boldsymbol{r}_h)\sqrt[\eta]{\frac{C_d}{2\left(E^{(ele)}+E^{(proc)}\right)}}$. Since h must be an integer, we define h^* to be the nearest integer to h that minimizes the energy. However, the constraint that $\alpha = \frac{\text{dist}(\boldsymbol{r},\text{Sink})}{h} \leq d_c$, that is $h \geq \left\lceil \frac{\text{dist}(\boldsymbol{r},\text{Sink})}{d_c} \right\rceil$ has to be satisfied.

1.3.1.4. *Computing* $F_{mE}(e|\boldsymbol{r})$

Given the probability P_a that a node is active, it is computed the cumulative probability $F_{mE}(e|\boldsymbol{r})$ that the minimum energy required to send a packet from a sensor to the sink, using only active nodes, is less than or equal to e, conditioned to the sensor being at $\boldsymbol{r}$. Given K_{Max} possible routes, the kth path (with $2 \leq k \leq K_{\text{Max}}$) will be used only if the next hop on route k is active and all previous $k-1$ routes are unavailable because the corresponding next hops are asleep.

Therefore, $F_{mE}(e|\boldsymbol{r})$ is written as:

$$F_{mE}(e|\boldsymbol{r}) = P_a\sum_{k=1}^{K_{\text{Max}}}(1-P_a)^{k-1}F_{mE}^{k}(e|\boldsymbol{r}). \qquad [1.13]$$

Note that $\lim_{e\to\infty} F_{mE}(e|\boldsymbol{r}) \leq 1$, and tends to one only if $K_{\text{Max}} \to \infty$. Since K_{Max} is assumed to be large enough so that the probability that none of the possible next hops are available is negligible, we have: $\lim_{e\to\infty} F_{mE}(e|\boldsymbol{r}) \approx 1$.

1.3.1.5. *Computing* $p_{s;\mathbf{r}}(\boldsymbol{r'}|e)$

$p_{s;\mathbf{r}}(\boldsymbol{r'}|e)$ is defined as the probability density that a sensor in $\boldsymbol{r}$ selects a sensor in position $\boldsymbol{r'}$ as next hop to send its packet to the sink with energy

expenditure e, conditioned to the fact that e is the minimum required energy. Clearly, $p_{s;\mathbf{r}}(\boldsymbol{r'}|e)$ is equal to 0 if the energy required to send a packet from position $\boldsymbol{r}$ to the sink (using as next hop a node in position $\boldsymbol{r'}$) is different from e. Otherwise, it can be computed based on the following observations: (1) the next hop is selected with uniform probability among all possible sensors that are suitable for the task, (2) the spatial density of active sensors in $\boldsymbol{r'}$ is $\rho(\boldsymbol{r'})P_a$, (3) the spatial density of active sensors that allow a packet to be transferred from $\boldsymbol{r}$ to the sink with energy cost equal to e is computed in this manner:

$$p_{s;\mathbf{r}}(\mathbf{r'}|e) = \frac{\rho(\mathbf{r'})\delta(\epsilon_m(\mathbf{r}, \mathbf{r'}) - e)}{\iint \rho(\mathbf{r''})\delta(\epsilon_m(\mathbf{r}, \mathbf{r''}) - e)d\mathbf{r''}} \quad [1.14]$$

where the Dirac delta function, $\delta(\cdot)$, ensures that $p_{s;\mathbf{r}}(\mathbf{r'}|e)$ is a proper probability distribution.

1.3.1.6. *Computing* $u(\boldsymbol{r'}|\boldsymbol{r})$

Finally, using [1.13] and [1.14], the routing function $u(\boldsymbol{r'}|\boldsymbol{r})$ is given by:

$$u(\boldsymbol{r'}|\boldsymbol{r}) = \int_e p_{s;\mathbf{r}}(\boldsymbol{r'}|e)\frac{\mathrm{d}F_{mE}(e|\boldsymbol{r})}{\mathrm{d}e}\,\mathrm{d}e \quad [1.15]$$

Indeed, the probability density that the next hop chosen by a sensor in $\boldsymbol{r}$ is located at $\boldsymbol{r'}$ can be obtained by deconditioning [1.14] with respect to the probability density that the minimum energy needed to send a packet from $\boldsymbol{r}$ to the sink (using only active nodes) is equal to e.

1.3.2. *Channel contention and data transmission: computing* $s(\boldsymbol{r})$ *and* $P_R(\boldsymbol{r})$

The channel access scheme described in section 3.1 is modeled as follows.

Consider a sensor in position $\boldsymbol{r}$ wishing to transmit a packet; the objective is to compute the average service time $s(\boldsymbol{r})$ and the average packet retransmission probability $P_R(\boldsymbol{r})$.

Since the packet service time depends on the traffic load over the wireless channel, one fundamental variable that we need to compute is the probability $P_B(\boldsymbol{r})$ that a sensor in $\boldsymbol{r}$ senses the channel as busy. Using this quantity, the average packet service time at a node in $\boldsymbol{r}$, $s(\boldsymbol{r})$ is expressed as

$$s(\boldsymbol{r}) = [1 - P_B(\boldsymbol{r})]s_I(\boldsymbol{r}) + P_B(\boldsymbol{r})s_B(\boldsymbol{r}) \quad [1.16]$$

where $s_B(\boldsymbol{r})$ ($s_I(\boldsymbol{r})$) is the average service time when a sensor in $\boldsymbol{r}$ observes a busy (idle) channel. Similarly, the average packet retransmission probability is given by:

$$P_R(\boldsymbol{r}) = [1 - P_B(\boldsymbol{r})]\, p_{cI}(\boldsymbol{r}) + P_B(\boldsymbol{r})\, p_{cB}(\boldsymbol{r}) \quad [1.17]$$

where $p_c^B(\boldsymbol{r})$ ($p_c^I(\boldsymbol{r})$) is the collision probability when a sensor in $\boldsymbol{r}$ observes a busy (idle) channel.

$P_B(\boldsymbol{r})$ can be derived as follows. The average probability that a sensor in $\boldsymbol{r'}$ occupies the channel can be written as the ratio of the actual traffic rate in $\boldsymbol{r'}$ to the available channel rate. It is assumed that the probability of finding the channel busy at one point in time is close to the time stationary probability. By denoting the available channel rate (expressed in packets/s) by B_w and considering all sensors within the carrier sensing range of the node at $\boldsymbol{r}$, it follows that:

$$P_B(\boldsymbol{r}) = \iint_{|\boldsymbol{r'}-\boldsymbol{r}|\leq ds} \frac{\Lambda\star(\boldsymbol{r'})}{Bw}\, \mathrm{d}\boldsymbol{r'} \quad [1.18]$$

The quantities $s_B(\boldsymbol{r})$ and $s_I(\boldsymbol{r})$ are computed as functions of the collision probabilities when the channel is busy/idle and parameters of the CSMA/CA protocol. The expression of $p_c^B(\boldsymbol{r})$ and $p_c^I(\boldsymbol{r})$ will be provided later; assume, for now, that these quantities are known.

For the sake of brevity, the sum of the duration of a data transmission, δ_a and the duration of an ACK transmission is denoted by L. Note that the duration of a data (ACK) transmission depends on the data (ACK) size and the available channel rate. The mean service time is derived by considering that the probability of multiple collisions is negligible [HEU 03]. To verify

such an assumption, simulations with $N = 400$ nodes have been run, in the case where all sensors are active. The integral over the network area of the traffic generation rate $\lambda_s(\boldsymbol{r})$ of the sensors has been set to 0.1, and the probability that a packet experiences one or more retransmissions has been evaluated (see section 1.4.1 for a description of the simulator). The results presented in Figure 1.1 show that the probability of having multiple collisions can indeed be neglected.

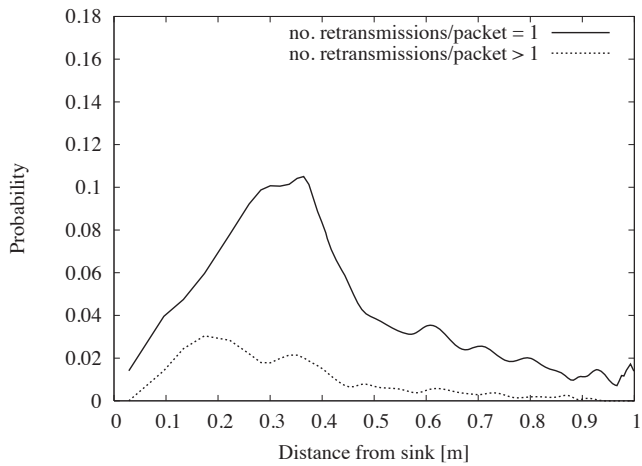

Figure 1.1. *Simulation results for the probability that a packet experiences one or more retransmissions, as a function of the node distance from the sink*

The mean service time when the channel is sensed idle can be expressed as

$$s_I(\boldsymbol{r}) = \left(1 - p_c^I(\boldsymbol{r})\right)(\delta_s + L) + p_c^I(\boldsymbol{r})\left(2\delta_s + 2L + \frac{CW}{2} + f \cdot L\right) \quad [1.19]$$

$$= (\delta_s + L)\left(1 + p_c^I(\boldsymbol{r})\right) + \left(\frac{CW}{2} + f \cdot L\right) p_c^I(\boldsymbol{r}) \quad [1.20]$$

where δ_s is the time interval during which a node senses the channel, CW is the CW size and f is the mean number of transmissions that make the backoff in $\boldsymbol{r}$ freeze (see section 3.1). The first term on the right-hand side of [1.19] accounts for the case where the node accessing the channel does not collide

with any other node. The second term represents the case where a collision occurs, the tagged node extracts a backoff value (the mean value of the backoff time is equal to $CW/2$ and the backoff is frozen for a time equal to $f \cdot L$) and the packet is retransmitted. The quantity f can be evaluated as $f = \frac{\overline{m}(\boldsymbol{r})}{2}$, where $\overline{m}(\boldsymbol{r})$ is the mean number of sensors in the proximity of $\boldsymbol{r}$ contending for the channel; the complete derivation of f is given in the following section. Because of the Poisson distribution of sensors in the area, $\overline{m}(\boldsymbol{r})$ can be computed as:

$$\overline{m}(\boldsymbol{r}) = \iint_{|\boldsymbol{r'}-\boldsymbol{r}|\leq ds} \frac{\Lambda\star(\boldsymbol{r'})}{Bw}\rho(\boldsymbol{r'})\,\mathrm{d}\boldsymbol{r'} \qquad [1.21]$$

Similarly, $s_B(\boldsymbol{r})$ is given by

$$s_B(\boldsymbol{r}) = \left(1 - p_c^B(\boldsymbol{r})\right)\left[\frac{L}{2} + \delta_s + \frac{CW}{2} + L(1+f)\right] + p_c^B(\boldsymbol{r})\left[\frac{L}{2} + \delta_s + \frac{CW}{2}\right.$$
$$\left. + L(1+f) + \delta_s + CW + +L(1+f)\right] \qquad [1.22]$$
$$= \left[\delta_s + \frac{CW}{2} + L(1+f)\right]\left(1 + p_c^B(\boldsymbol{r})\right) + \frac{CW}{2}p_c^B(\boldsymbol{r}) + \frac{L}{2} \qquad [1.23]$$

Equation [1.22] can be explained as follows. Since the channel is sensed to be busy, first the node has to wait for the current transmission to end, i.e. on average a time interval equal to $L/2$; then, after a time δ_s, it extracts the backoff time (whose average value is equal to $CW/2$). Again, during the backoff procedure other nodes may access the channel thus making the backoff freeze (for an average time duration of $f \cdot L$). If a collision takes place, an additional backoff procedure has to be performed and the packet has to be retransmitted. Note that, at the second backoff extraction, the value of CW is doubled, thus the average value of backoff to be considered is equal to CW.

In order to compute the expressions above, the following equations are applied: $p_c^I(\boldsymbol{r})$ and $p_c^B(\boldsymbol{r})$:

$$p_c^I(\boldsymbol{r}) = \iint u(\boldsymbol{r'}|\boldsymbol{r}) \iint_{|\boldsymbol{r''}-\boldsymbol{r'}|\leq d_s} \frac{\Lambda^\star(\boldsymbol{r''})}{B_w}\,\mathrm{d}\boldsymbol{r''}\,\mathrm{d}\boldsymbol{r'} \qquad [1.24]$$

Equation [1.24] is derived considering that, when the channel is idle, a node's transmission collides if any other node located within the carrier sensing range of the receiver is transmitting. In [1.24], $u(\boldsymbol{r'}|\boldsymbol{r})$ is the probability that the next hop of the transmitter in $\boldsymbol{r}$ is located in $\boldsymbol{r'}$, while the inner integrals represent the probability that the next hop (i.e. the receiver) observes a busy channel.

When the channel is busy, a collision occurs if two or more nodes within the radio range of the receiver extract the same backoff value, thus accessing the channel at the same time. Because of the fact that multiple collisions are neglected, it can be assumed that all nodes set the size of their CW to the same value CW. Thus, given that $1/CW$ is the probability that a contending sensor selects the same backoff as the tagged node, it is fair to write:

$$p_c^B(\boldsymbol{r}) = \iint u(\boldsymbol{r'}|\boldsymbol{r}) \sum_{k=0}^{\infty} \left[1 - \left(1 - \frac{1}{CW}\right)^k\right] \frac{e^{-\overline{m}(\boldsymbol{r'})}(\overline{m}(\boldsymbol{r'}))^k}{k!} \mathrm{d}\boldsymbol{r'} \quad [1.25]$$

Again, $u(\boldsymbol{r'}|\boldsymbol{r})$ accounts for the probability that the tagged sensor (located in $\boldsymbol{r}$) uses the node in $\boldsymbol{r'}$ as next hop, while $\overline{m}(\boldsymbol{r'})$ is the average number of sensors in the proximity of $\boldsymbol{r'}$ contending for the channel.

The average probability to retransmit a packet at $\boldsymbol{r}$, $P_R(\boldsymbol{r})$, can be computed using [1.17]. Finally, to be consistent with the assumption that packets are retransmitted at most once, [1.5] can be rewritten as,

$$\Lambda\star(\boldsymbol{r}) = \Lambda(\boldsymbol{r})[1 + P_R(\boldsymbol{r})], \quad [1.26]$$

which provides the actual traffic rate density at point $\boldsymbol{r}$.

Equations [1.17], [1.18], [1.21] and [1.24]–[1.26] all depend on each other; therefore, these equations are solved by means of a fixed point approximation (FPA) procedure. At the first iteration of the FPA procedure, $\Lambda^\star(\boldsymbol{r})$ is considered to be equal to $\Lambda(\boldsymbol{r})$, and [1.17], [1.18], [1.21], [1.24] and [1.25] are solved in the same order as listed here. Then, [1.26] is used to obtain a new value of $\Lambda^\star(\boldsymbol{r})$, and then the procedure is repeated until convergence on the parameter estimates is reached. Note that very few iterations are needed to reach convergence (namely, 3 and 4).

1.3.2.1. *Computation of the mean number of transmissions freezing the backoff counter*

The probability that a sensor draws one out of the CW backoff values is $1/CW$. Now, consider that a node has drawn the backoff value j, with $0 \leq j < CW$. The probability that the decrease in the node backoff counter is interrupted by i transmissions is the probability that i out of the k contending nodes have selected a smaller backoff value, i.e. $j/(CW-1)$. The number of contending nodes k is distributed according to a Poisson distribution with mean $\overline{m}(\boldsymbol{r})$. It follows that the average number of transmissions that make the backoff counter of the sensor freeze can be computed as

$$f = \frac{1}{CW}\sum_{k=0}^{\infty}\frac{e^{-\overline{m}(\boldsymbol{r})}(\overline{m}(\boldsymbol{r}))^k}{k!}\sum_{j=0}^{CW-1}\sum_{i=0}^{k} i\binom{k}{i}\left(\frac{j}{CW-1}\right)^i \times \left(\frac{CW-j-1}{CW-1}\right)^k = \sum_{k=0}^{\infty}\frac{e^{-\overline{m}(\boldsymbol{r})}(\overline{m}(\boldsymbol{r}))^k}{k!}k\frac{\sum_{j=0}^{CW-1} j}{CW(CW-1)}$$

$$= \frac{1}{2}\sum_{k=0}^{\infty} k\frac{e^{-\overline{m}(\boldsymbol{r})}(\overline{m}(\boldsymbol{r}))^k}{k!} = \frac{\overline{m}(\boldsymbol{r})}{2}$$

1.3.3. *Mean packet delivery delay: computing* $q(\boldsymbol{r})$

In section 1.1, the mean packet delivery delay and [1.6] are derived to compute this quantity; the equation contains one term representing the average queueing delay experienced by a packet at point $\boldsymbol{r}$, denoted by $q(\boldsymbol{r})$ whose computation is shown here. First, for the sake of clarity, the simple case where sensors are always active is considered, then the general case where sensors alternate between active and sleep modes is addressed.

1.3.3.1. *Computing* $q(\boldsymbol{r})$ *for always active sensors*

The service time defined in section 1.3.2 can be used to derive an approximate expression for the average queueing delay experienced by a packet at each sensor it goes through. To this end, the queue at each sensor is modeled by using a simple M/M/1 queueing system where the arrival rate is defined by [1.4], while the average service time is defined by [1.16]. It is important to point out that this is an approximation of the actual behavior of

the system since the arrival as well as the service processes of data packets at a sensor are not exponentially distributed. The use of this simple model is justified by its simplicity and by its accuracy, as compared to simulation results. Moreover, it allows an easy extension to the case of finite buffer size, considering the $M/M/1/B$ queue model.

Exploiting well-known results from the M/M/1 queue theory, the mean queue length experienced by a packet at point $\boldsymbol{r}$ can be computed as:

$$\overline{n}(\boldsymbol{r}) = \frac{\Lambda(\boldsymbol{r})s(\boldsymbol{r})}{1 - \Lambda(\boldsymbol{r})s(\boldsymbol{r})} \qquad [1.27]$$

and by using Little's law, the average queueing delay is given by:

$$q(\boldsymbol{r}) = \frac{\overline{n}(\boldsymbol{r})}{\Lambda(\boldsymbol{r})} = \frac{s(\boldsymbol{r})}{1 - \Lambda(\boldsymbol{r})s(\boldsymbol{r})} \qquad [1.28]$$

1.3.3.2. *Computing $q(\boldsymbol{r})$ for active and sleeping sensors*

Let us consider now that sensors may be either in *active* or in *sleep* state. While being asleep, a sensor cannot generate, receive or transmit data packets but it preserves its buffer data content. To derive an expression for the average queueing delay experienced by a packet at point $\boldsymbol{r}$ in case of active/sleep dynamics, the behavior of a sensor is modeled by means of a birth/death model that suspends the arrivals as well as the services when the sensor is in sleep mode. In terms of queuing theory, these assumptions correspond to considering an M/M/1 queue with server vacations and interrupted arrivals (in the following, ON and OFF are used as shorthands to indicate active and sleep states of the server).

Figure 1.2 shows a possible evolution of the number of queued packets at a node. Notice that, during the OFF period, the number of packets in the queue is frozen, because both arrivals and services are suspended. This behavior corresponds to a special case of queueing system with vacations [DOS 86] that allows for a very simple analysis. Indeed, if the OFF periods are removed, the same dynamics would occur as in the case in which nodes are always active (see section 1.3.3.1). Moreover, OFF periods start at random points in time, with no correlation with the state of the queue. As a result, at any time the number of packets in the queue has the same distribution as the

stationary distribution derived in the case in which nodes are always on. Note that this property holds in general, i.e. for any distribution of the duration of ON and OFF periods.

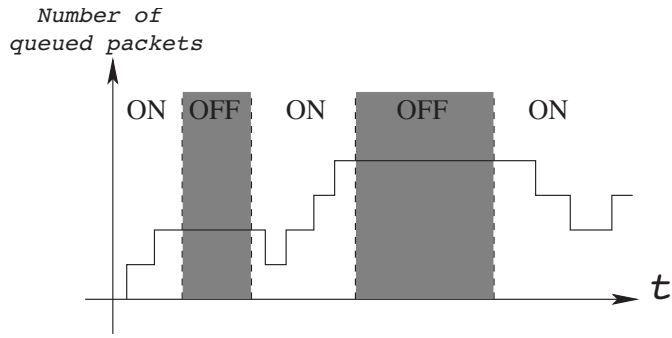

Figure 1.2. *Example of evolution of the number of queued packets at a node in the case of active-sleep dynamics*

Given that the aim is to derive the average time spent in the queue, the application of Little's law produces

$$q(\boldsymbol{r}) = \frac{\overline{n}(\boldsymbol{r})}{\Lambda(\boldsymbol{r})\, Pa} \qquad [1.29]$$

where $\overline{n}$ is the same as in [1.27], whereas $P_a = \bar{T}_{\mathrm{ON}}/(\bar{T}_{\mathrm{ON}} + \bar{T}_{\mathrm{OFF}})$ is the probability that a sensor is active, which depends only on the average durations $\bar{T}_{\mathrm{ON}}$ and $\bar{T}_{\mathrm{OFF}}$ of the ON and OFF phases, respectively.

1.4. Results

This section presents a collection of results obtained exploring the parameter space of the network scenario described in section 3.1. In particular, in section 1.4.1 the analytical predictions derived from the fluid model are compared with those obtained using a detailed simulator of the sensor network in the case of homogeneous node deployment. In section 1.4.2, the model is exploited to study the behavior of large-scale sensor networks as the active/sleep dynamics and the sensing/communication range of the nodes vary. Furthermore, the impact of non-homogeneous node

deployment is investigated assuming that the sensor density varies according to a (truncated) exponential distribution as a function of the distance from the sink.

1.4.1. *Model validation*

The development of an *ad hoc* discrete-time simulator allowed for the validation of the analytical results. The simulator is based on the assumptions about the system behavior specified in section 3.1. At the beginning of the simulation, a random (connected) topology is generated using a uniform distribution of nodes over a disk area of unit radius. Then, it is computed once and for all the minimum-energy next hops available to each node, according to the energy cost defined in section 3.1. The simulator implements all details of the CSMA/CA access mechanism considered in this chapter that resembles the operation of the IEEE 802.11 DCF. The duration of a slot is $\sigma = 320$ μs [IEE 06], δ_s is equal to 50 μs and the CW is CW=16. The available channel rate is B_w=250 kbit/s and the packet size (including the packet header) is equal to 400 bits; L results to be 1.92 ms. Data packets are generated by active sensors according to a Poisson process.

Since the simulation results are obtained for a particular instance of sensors deployment, the correct validation methodology requires averaging the simulation predictions over a large number of deployment realizations. The simulation process has run 200 experiments for each set of system parameters, and the average and the 90% confidence intervals have been computed for each performance metric. Each simulation experiment discards an initial transient period that we set equal to $5,000$ time slots. The length of each run is equal to $200,000$ time slots. Furthermore, while the model allows for the computation of the spatial distribution of performance metrics, to ease the interpretation of results as well as the comparison against simulation outcomes, the simulator averages the performance metrics over the space points at the same distance from the sink. The result is given in 2D graphs of the performance metrics as functions of the distance from the sink. In all the plots shown in this section, confidence intervals are depicted as vertical error bars.

The values for the system parameters are reported in the following. The number of sensors in the network is $N = 400$. The communication range is equal to the carrier sensing range for all nodes, $d_c = d_s = d = 0.25$. The

energy consumption parameters are set to: $C_d = 0.018$ mJ, $E^{(\mathrm{ele})} = E^{(\mathrm{proc})} = 0.15$ mJ, and $P_i = 18$ mW [MIC 04]. Let us denote by L_g the integral over the network area of the traffic generation rates $\lambda_s(\boldsymbol{r})$ of the sensors. To validate the model under different system loads, two values of L_g have been set: 0.1 and 0.15. Note that the system load is considered normalized to the available channel rate. Furthermore, with this set of parameters the system is not stable for values of L_g greater than 0.2.

The traffic rate density, $\Lambda(\boldsymbol{r})$, which accounts for both packets generated by the node and the relay traffic, is the first metric evaluated. Figure 1.3 shows the behavior of $\Lambda(\boldsymbol{r})$ versus the node distance from the sink. Note that the traffic rate (and, hence, power consumption) is not evenly distributed across the network; indeed, nodes closer to the sink have to relay a larger amount of traffic than peripheral sensors. This is related to the well-known problem of data implosion at the sink [KUL 02] that affects multipoint-to-point communications and results in unfairness among the network nodes. It is also important to observe that traffic rate per sensor diminishes significantly at distances close to multiples of the communication range d. This is due to the routing strategy adopted, which selects the path minimizing the overall energy cost to send a packet to the sink. Notice that, due to the choice of values for C_d, $E^{(\mathrm{ele})}$ and $E^{(\mathrm{proc})}$, it turns out that the fixed cost required to transmit a packet over one hop is much larger than the variable cost due to the amplifier, even if sending at the maximum possible distance d. As a result, routes are primarily selected on the basis of the minimum hop count. Among all routes having the minimum number of hops, the one minimizing the variable cost due to the amplifier is preferred. Finally, Figure 1.3 highlights that the model is able to accurately predict the particular shape of the curves obtained for different values of L_g.

The graphs on the left-hand side Figures 1.4 and 1.5 show the average data delivery delay versus the source distance from the sink (i.e. the time elapsed from the instant at which a packet is generated by a sensor to the instant at which the packet reaches the sink), for L_g equal to 0.1 and 0.15, respectively. The delay increases with the distance from the sink as well as with the value of L_g, as expected. The delivery delay is very well approximated by a model that represents the behavior of each node by a simple M/M/1 queue. The matching is pretty good for all considered values of L_g. The graphs on the right-hand side of Figures 1.4 and 1.5 present the power consumption per node, as a function of the node distance from the sink, for L_g equal to 0.1 and 0.15, respectively. Note

that, looking at the value of the various contributions to power consumption, it turns out that the one due to overhearing dominates. This suggests that, in order to save energy, it is of fundamental importance to build sensors that promptly detect when data packets are not destined to themselves and disregard them.

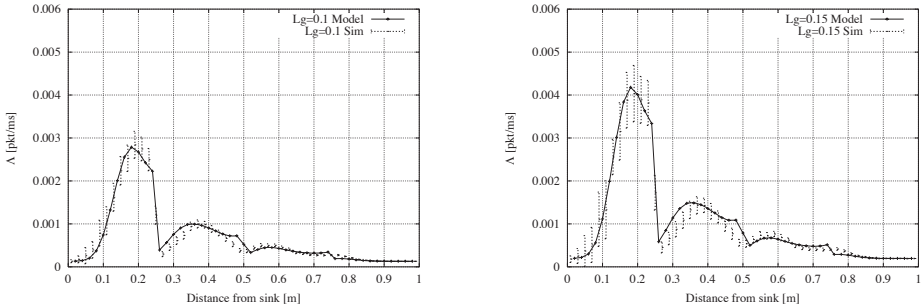

Figure 1.3. *Average traffic rate per sensor, Λ, versus the node distance from the sink, for $L_g = 0.1$ (left) and $L_g = 0.15$ (right). Analytical and simulation results are compared*

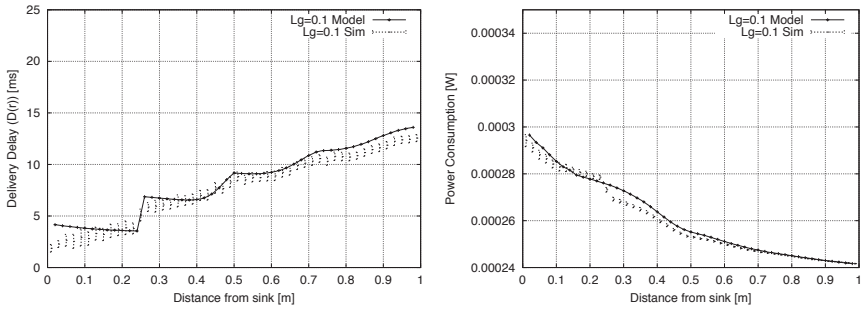

Figure 1.4. *Analytical and simulation results for $L_g = 0.1$ of the average delivery delay (left) and of the average power consumption (right) as functions of the node distance from the sink*

The model validation for the case sensors exhibiting active/sleep behavior is reported in Figure 1.6. It shows the packet delivery delay (top graph) and the power consumption per node (bottom graph) as functions of the node distance from the sink. Both graphs present results for a scenario where sensors are in the sleep state for 50% of the time and for $L_g = 0.1$. According to the model assumptions, when a sensor enters sleep mode, its sleep time (expressed in slots) is geometrically distributed with parameter equal to 0.01; when the

sensor switches to the active mode, the scheduled active period (expressed in slots) is a random variable geometrically distributed with parameter equal to 0.01.

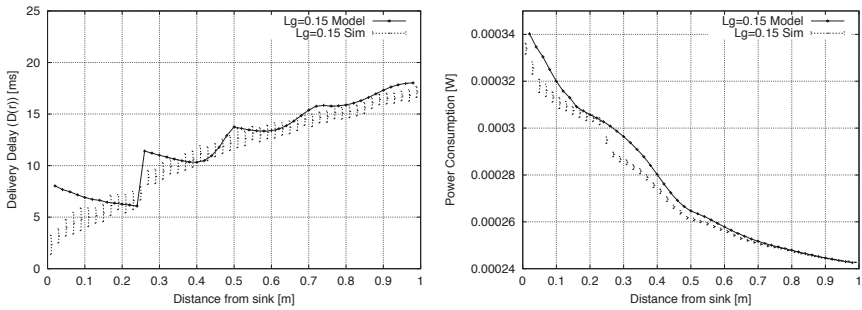

Figure 1.5. *Analytical and simulation results for $L_g = 0.15$ of the average delivery delay (left) and of the average power consumption (right) as functions of the node distance from the sink*

It is worth pointing out that, again, the data delivery delay is well approximated by our M/M/1 model with server vacations and interrupted arrivals, whose parameters are derived from the traffic rate and the service rate defined in section 1.3.2. For the sake of comparison, the results for the case where sensors are always active are reported in the two graphs of Figure 1.6.

In this manner, it is possible to appreciate the effect of the active/sleep dynamics on the delivery delay as well as on the power consumption. Indeed, the data delivery delay approximately doubles with respect to the case in which nodes are always active, as predicted by [1.29], since here $P_a = 0.5$.

1.4.2. *Model exploitation*

The application of the fluid-based model allows for the study of many issues relevant to sensor networks and the exploitation of different architectural solutions and parameter settings. Because of the detailed modeling approach, a large number of different system parameters can be investigated (e.g. number of sensors, carrier sensing/communication ranges, fraction of time in active/sleep mode). The results are obtained with a low computational solution cost and good accuracy.

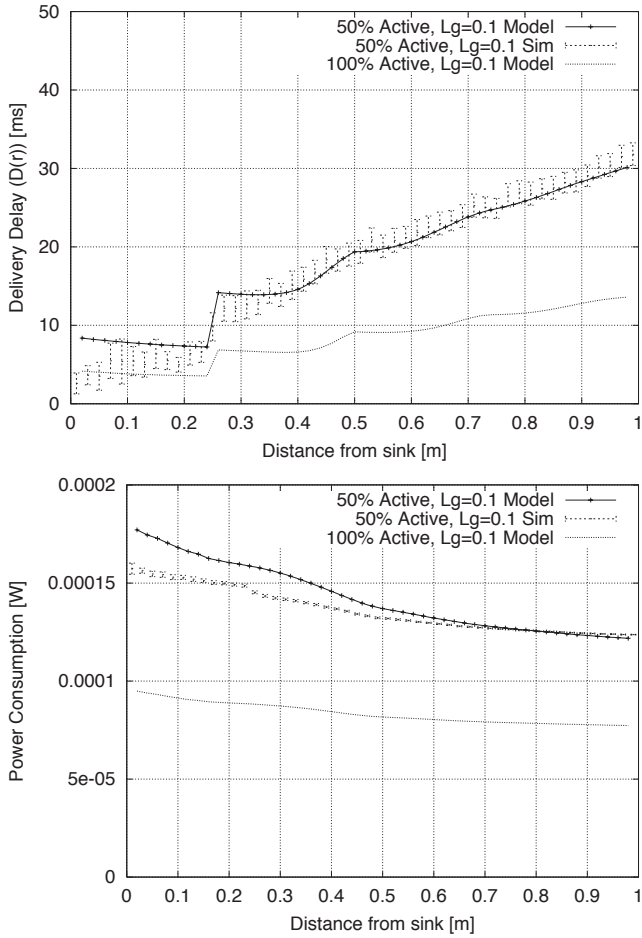

Figure 1.6. *Analytical and simulation results for the average delivery delay (top) and the average power consumption (bottom) as functions of the node distance from the sink, for $L_g = 0.1$. The results obtained when sensors are in the sleep state for 50% of the time are compared to the case where sensors are always active*

Figure 1.7 presents some results that allow us to appreciate the effect of different values for the common carrier sensing/communication range d (i.e. when $d_c = d_s = d$). Sensors are assumed to be always active. The scenario under investigation is characterized by the following parameters: $L_g = 0.1$, $N = 1,000$ and $d = 0.15, 0.1, 0.05$. The figure shows the average delivery

delay (top graph) and power consumption (bottom graph) as functions of the node distance from the sink: on the one hand for smaller values of d, the number of hops from the sensors to the sink grows, thus leading to a significant increase in the delivery delay; on the other hand, large values of d imply that distant transmissions also allow a node detect the channel as busy, therefore sensors often overhear other nodes' traffic and their power consumption increases.

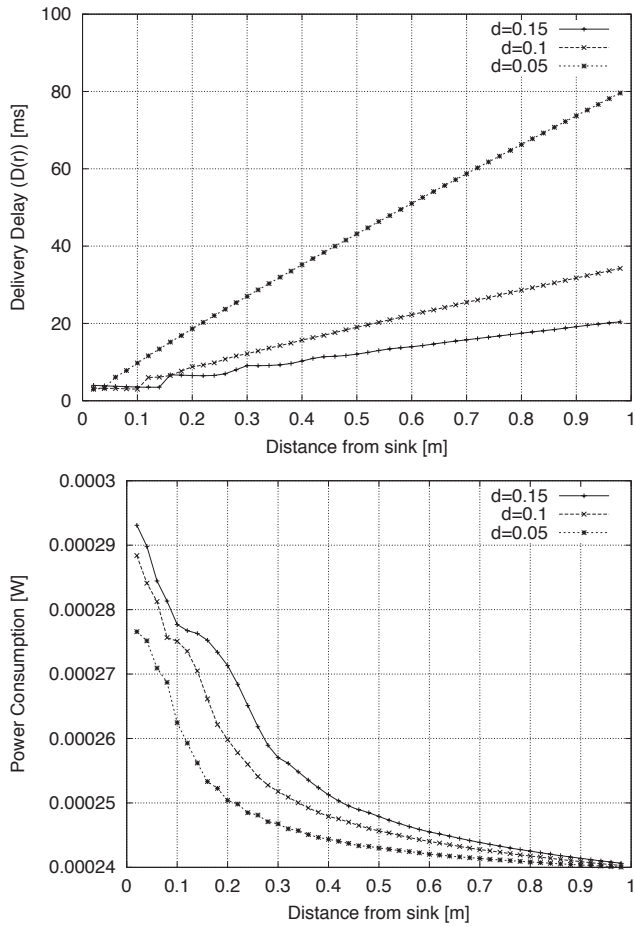

Figure 1.7. *Analytical results for the scenario with $L_g = 0.1$, $N = 1,000$, and $d = 0.15, 0.1, 0.05$. Average delivery delay (top) and average power consumption (bottom plot) as functions of the node distance from the sink*

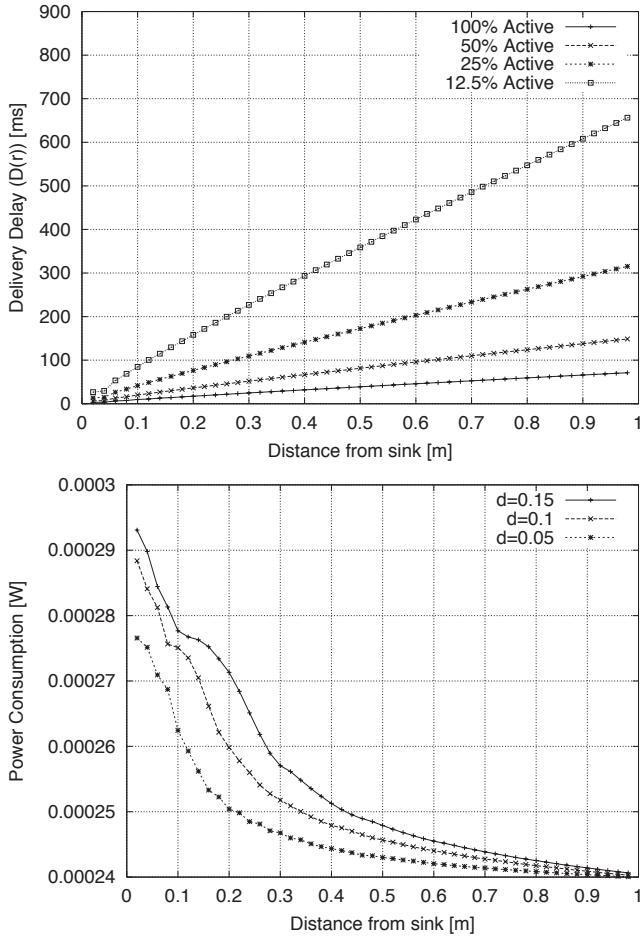

Figure 1.8. *Analytical results for the scenario with $L_g = 0.1$, $N = 5,000$, $d = 0.05$, and different fractions of time in active state (from 100 to 12.5%). Average delivery delay (top) and average power consumption (bottom) as functions of the node distance from the sink*

Figure 1.8 shows some results that allow us to appreciate the effect of different fractions of time spent in active mode (from 100 to 12.5%). The scenario being investigated is characterized by the following parameters: $L_g = 0.1$, $N = 5,000$ and $d = 0.05$. The figure presents the average delivery delay (top graph) and the average power consumption (bottom graph) as

functions of the node distance from the sink. Observe that the node power consumption significantly decreases as sensors spend more time in a sleep state, however this benefit has a cost in terms of delivery delay. Thus, a clear trade-off emerges between power consumption and the level of service that the network can offer.

The model has been also applied to a non-homogeneous scenario where the sensor density varies across the disk of unit radius: sensor deployment which the local density varies as a function of the distance from the sink according to a (truncated) exponential distribution of parameter α. In this case, by properly normalizing the distribution so that the average number of sensors N is kept constant, the sensor density is expressed as follows

$$\rho(\boldsymbol{r}) = \frac{N}{2\pi} \frac{\alpha^2 \ \exp(\alpha \cdot \mathrm{dist}(\boldsymbol{r}, \mathbf{Sink}))}{(\alpha - 1)e^{\alpha} + 1}$$

Figure 1.9 shows the sensor density for different values of the control parameter α. When $\alpha < 0$, the sensor density decreases as the distance from the sink increases, whereas for $\alpha > 0$ the sensor density increases while moving away from the sink. Notice that $\alpha = 0$ corresponds to the case of homogeneous sensor density, i.e. $\rho(\boldsymbol{r}) = N/\pi$.

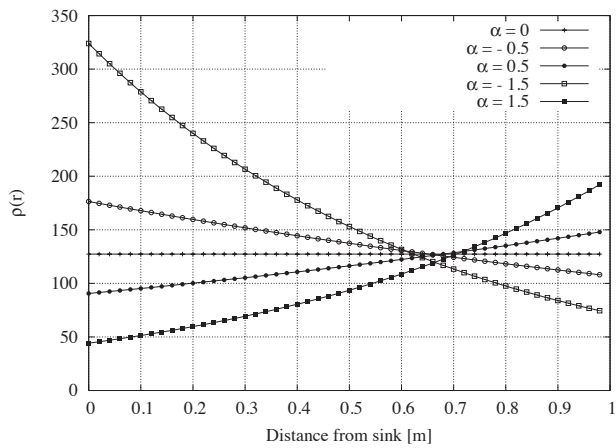

Figure 1.9. *Sensor density for different values of α as a function of the distance from the sink*

The following figures consider scenarios with $N = 400$, where $L_g = 0.1$, $d = 0.25$, and $N = 1,000$, where $L_g = 0.1$ and $d = 0.15$ or 0.05. Sensors are assumed to be always active.

Figure 1.10 reports the comparison of the average traffic rate per sensor for different values of the control parameter α in the case in which $N = 400$ (the case of $N = 1,000$ leads to similar results). Note that, for negative values of α, the values of $\Lambda(r)$ are smaller than in the homogeneous case ($\alpha = 0$), since the amount of traffic that has to be relayed over multiple hops (such as that produced by far away sensors) decreases. Moreover, for negative α the distribution of $\Lambda(r)$ is more even, leading to an improved fairness among sensors. The opposite is true for $\alpha > 0$.

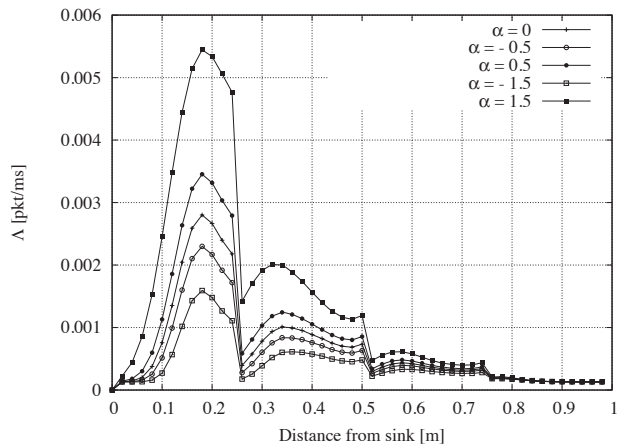

Figure 1.10. *Average traffic rate per sensor, Λ, in the case of $N = 400$ and different values of α*

Figure 1.11 shows the average delivery delay $D(r)$ for $N = 400$ (top) and $N = 1,000$ (bottom). For $N = 400$, negative values of α result into a larger delivery delay with respect to the case of homogeneous density. In particular, when $\alpha = -1.5$ the delay is significantly higher than the one obtained for the other considered values of α. This can be explained by the fact that, for negative α, even if the traffic load per sensor is lower (see Figure 1.10), the total number of sensors within the carrier sensing/communication range of the sink is much larger, resulting in an increased channel contention around the sink. In particular, for $\alpha = -1.5$, the wireless channel turns out to be very

congested near the sink (i.e. the local load approaches the maximum channel capacity) and queueing delays become very large.

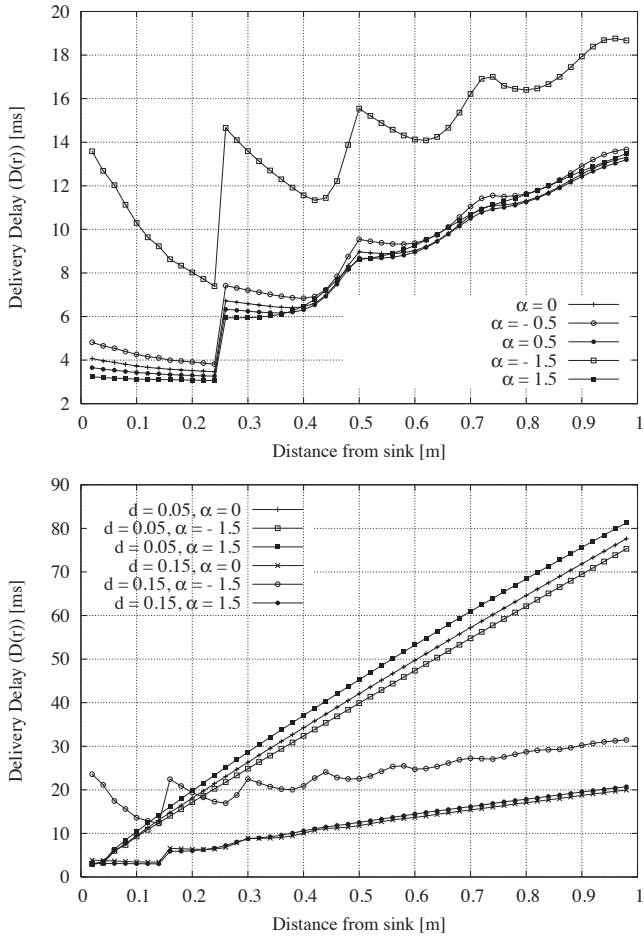

Figure 1.11. *Average delivery delay for $N = 400$ (top plot) and for $N = 1,000$ (bottom plot) as functions of the node distance from the sink and for different values of α*

At the other extreme, $\alpha = 1.5$ results in smaller delivery delays for sensors close to the sink, because in this case the channel contention around the sink is very low. However, the average delay of sensors far from the sink become slightly higher with respect to the case of homogeneous density; again, this is

due to the large sensor density and, thus, to the increased channel contention in the regions close to the network border.

A similar behavior is observed in the case of $N = 1,000$ and $d = 0.15$, as shown in Figure 1.11 (bottom). However, if we reduce the carrier sensing/communication range in this scenario to as small as $d = 0.05$, we observe a quite different behavior. When $d = 0.05$, a short distance is covered by each transmission, hence several hops are needed to far away sensors to deliver their traffic to the sink. As a result, delivery delays are small close to the sink, but get larger and larger while moving toward the network border (almost linearly as the distance from the sink increases), due to the increasing number of hops required to reach the sink. In this scenario, with $\alpha = -1.5$ the channel spatial reuse is large enough such that the channel contention around a node is always very low, and no congestion arises around the sink. Therefore, delivery delays are smaller than in the case of homogeneous density. For $\alpha = 1.5$, instead, few sensors are located around the sink and all of them are heavily used as relays by further away nodes, thus nodes experience a higher delivery delay than for $\alpha = 0$.

1.4.3. *Model solution complexity and accuracy*

The solution algorithm discretizes the disk of unit radius using N_p points, thus obtaining a radial discretization step equal to $\frac{1}{\sqrt{N_p}}$. The computational cost of the model solution is very limited and does not depend on the number of sensors. The control processing unit (CPU) time required to solve the model and obtain the results presented in section 1.4.1 is only a few seconds using $N_p = 2,500$, while the simulation experiments took much longer to complete. The simulation time increases more than linearly with the number of sensors, thus allowing to simulate at most a few hundreds of sensors.

Furthermore, increasing the number of discretization points results in a slightly increased accuracy of the model predictions at the cost of an increased CPU time required for the model solution. Figure 1.12 displays the model predictions for the average traffic rate per sensor under the same scenario as in the left plot in Figure 1.3, for $N_p = 400, 1600, 2500, 10000$. It can be observed that both $N_p = 2,500$ and $N_p = 10,000$ yield very accurate results, although the model solution time is significantly larger in the latter

case (tens of seconds). Therefore, $N_p = 2,500$ turns out to be a reasonable trade-off between efficiency and accuracy of the discretization algorithm.

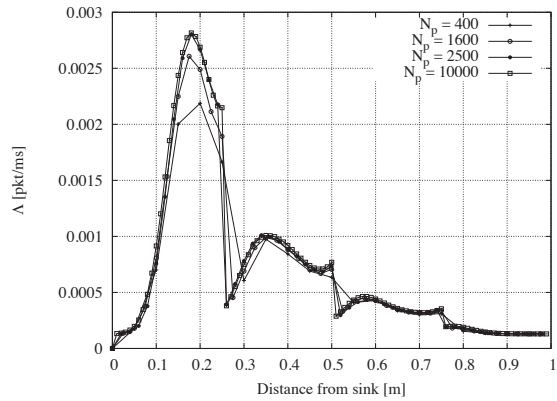

Figure 1.12. *The effect of the number of discretization points in the solution algorithm on the accuracy of the model predictions*