Algorithmic Models for Sensor Networks*

Stefan Schmid[1], Roger Wattenhofer[2]

[1]ETH Zurich
Computer Engineering and Networks Laboratory
CH-8092 Zurich, Switzerland
schmiste@tik.ee.ethz.ch

[2]ETH Zurich
Computer Engineering and Networks Laboratory
CH-8092 Zurich, Switzerland
wattenhofer@tik.ee.ethz.ch

Abstract

Developing algorithms for sensor networks—and proving their correctness and performance—, requires simplifying but still realistic models. This paper surveys various models in use today and puts them into perspective. In addition, we propose interesting models which are not widely adopted by the community so far.

1 Introduction

Sensor networks are used to gather distributed information from a given location or region. These networks consist of tiny devices which are equipped with a power source, a microprocessor, a wireless interface, some memory, and one or more sensors. The sensors are used for collecting physical parameters such as light intensity, sound, or temperature. Because of the limited radio communication range, sensor nodes communicate (e.g., gather data) by wireless multi-hop routing via intermediate nodes.

In order to develop algorithms for sensor networks and in order to give mathematical proofs of their correctness and performance, appropriate models are needed. Finding good models however is a challenging task. On the one hand, a model should be as simple as possible such that the analysis of a given algorithm remains tractable. On the other hand, however, a model must not be too simplistic in the sense that it neglects important properties of the network. A great algorithm in theory may be inefficient or even incorrect in practice if the analysis is based on idealistic assumptions. For example, an algorithm which ignores interference may fail in practice since communication happens over a shared medium.

Many models for sensor network have their origin in classic areas of theoretical computer science and applied mathematics. Since the topology of a sensor network can be regarded as a *graph*, the distributed algorithms community uses models from *graph theory*, representing nodes by vertices and wireless links by edges. Another crucial ingredient of sensor network models is *geometry*. Geometry comes into play as the distribution of nodes in space, and the propagation range of wireless links, usually adhere to geometric constraints.

Of course, a good model depends on the question studied. A media access study might need a detailed model capturing several low-level aspects. For instance, it has to be taken into account that a message might not be received correctly due to a near-by concurrent transmission. Hence, it is crucial that the model appropriately incorporates interference aspects. However, for a transport layer study, a much simpler model which assumes random transmission errors might be sufficient.

This paper discusses various models for sensor networks and strives at putting them into perspective. The selection of models presented is of course far from being complete and also highly subjective. In particular, our focus is on models of higher levels of abstraction; a large body of interesting work about the physical layer (e.g., cf. [6]) is not considered.

We hope that the paper can provide insights into how sensor network models are related, and that it can give advice as to when and why which model should be preferred.

The paper is organized as follows. In Section 2, models are studied for the network's connectivity. Section 3 then enhances these models by adding interference aspects: Since the communication medium is shared, sensor nodes might not be able to communicate although they are within each other's transmission range. Finally, Section 4 compares the different kinds of sensor network algorithms in use today, and discusses several modeling choices which influence the design and performance of an algorithm. The paper is concluded in Section 5.

*Research supported by the Swiss National Science Foundation.

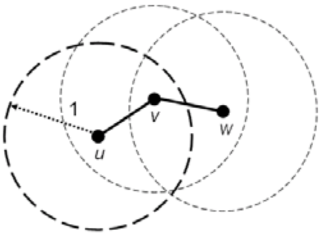

Figure 1. Unit disk graph: node u is adjacent to node v (distance ≤ 1), but not to node w (distance > 1).

2 Connectivity

A first and foremost modeling question concerns the connectivity of sensor nodes: Given a set of nodes distributed in space, we need to specify which nodes can receive a transmission of a node. Throughout this paper, if a node u is within a node v's transmission range, we say that u is *adjacent* to v, or equivalently, that u is a *neighbor* of v. In the absence of interference (cf. Section 3), this relation is typically *symmetric* (or *undirected*), i.e., if a node u can hear a node v, also v can hear u.

The classic connectivity model is the so-called *unit disk graph* (UDG) [3].[1] Nodes having omnidirectional radio antennas are assumed to be deployed in a planar, unobstructed environment. Two nodes are adjacent if and only if they are within each other's transmission range (which is normalized to 1).

Model 2.1 (Unit Disk Graph (UDG)) *Let $V \subset \mathbb{R}^2$ be a set of nodes in the 2-dimensional plane. The Euclidean graph $G = (V, E)$ is called* unit disk graph *if any two nodes are adjacent if and only if their Euclidean distance is at most 1. That is, for arbitrary $u, v \in V$, it holds that $\{u, v\} \in E \Leftrightarrow |u, v| \leq 1$. Figure 1 depicts an example of a UDG.*

The UDG model is quite idealistic: in reality, radios are not omnidirectional, and even small obstacles such as plants could change connectivity. For these reasons, some researchers have proposed to study the other extreme and model the sensor network as a *general graph*.

Model 2.2 (General Graph (GG)) *The connectivity graph is a* general undirected graph *G.*

While a UDG is too optimistic, the GG is often too pessimistic, as the connectivity of most networks is not arbitrary but obeys certain geometric constraints. Still, in some application scenarios it might be accurate to operate either on the UDG or on the GG. Indeed, there are algorithms developed for the UDG which also perform well in more general models. Moreover, some algorithms designed for the GG are currently also the most efficient for UDGs (e.g. [15]).

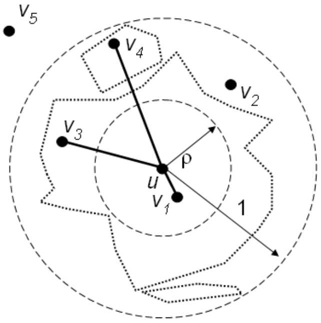

Figure 2. Quasi unit disk graph from the perspective of node u: Node u is always adjacent to node v_1 ($d(u, v_1) \leq \rho$) but never to v_5 ($d(u, v_5) > 1$). All other nodes may or may not be in u's transmission range. In this example, node u is adjacent to v_3 and v_4, but not to v_2.

The research community has inevitably searched for connectivity models between the two extremes UDG and GG. For example, the *quasi unit disk graph model* (QUDG) [1, 16] is a generalization of the UDG which takes imperfections into account as they may arise from non-omnidirectional antennas or small obstacles.[2]

Model 2.3 (Quasi Unit Disk Graph (QUDG)) *The nodes are in arbitrary positions in $\mathbb{R}^2$. All pairs of nodes with Euclidean distance at most ρ for some given $\rho \in (0, 1]$ are adjacent. Pairs with a distance larger than 1 are never in each other's transmission range. Finally, pairs with a distance between ρ and 1 may or may not be neighboring. An example is shown in Figure 2.*

Fact 2.4 *A QUDG with $\rho = 1$ is a UDG.*

Option 2.5 (QUDG Variations) *The QUDG as presented in Model 2.3 does not specify precisely what happens if the distance is between ρ and 1. There are several options. For example, one could imagine an* adversary *choosing for each node pair whether they are in each other's transmission range or not. Alternatively, there may be a certain* success probability *of being adjacent: the corresponding probability distribution could depend on the time and/or distance [25]. For example, the QUDG could be used to study*

[1] The name "unit disk graph" stems from the area of *computational geometry*. The UDG is a special case of an *intersection graph*.

[2] Note that QUDGs are related to *civilized graphs* [10].

Figure 3. Nodes u and v are separated by a wall. Nodes on the same side of the wall are completely connected. However, due to the wall, although u can reach a distant node w, it cannot hear the close node v. Such situations can be modeled by the BIG, but not by the UDG or the QUDG.

Options 2.8 (Generalized (Q)UDG) *One extension of the UDG and QUDG models is to consider nodes in $\mathbb{R}^3$. Moreover, distances between nodes could be modeled using the Manhattan norm ($\mathbb{L}_1$ norm) or the maximum norm ($\mathbb{L}_\infty$ norm) rather than assuming Euclidean distances ($\mathbb{L}_2$ norm).*

For instance in [13], the UDG definition has been extended to *doubling metrics* [8].[4]

Model 2.9 (Unit Ball Graph (UBG)) *A doubling metric space is defined as follows: For a node u, let the ball $B_u(r)$ denote the set of all nodes at a distance at most r from u. It holds, for all nodes u and all $r \geq 0$, that the ball $B_u(r)$ can be covered by a constant number of balls of radius $r/2$, i.e., $B_u(r) \subseteq \bigcup_{i=1\ldots c} B_{u_i}(r/2)$, where u_i are arbitrary nodes and c is a (usually small) constant. In the UBG model, nodes are assumed to form a doubling metric space. Two nodes u and v with $d(u,v) \leq 1$ are adjacent, all other nodes are not.*

Facts 2.10 *Nodes in a 2-dimensional Euclidean plane (i.e., the metric space is given by the Euclidean distances), form a doubling metric. A general graph however does not.*

Figure 4 shows an example for the Euclidean plane. To see why a general graph may not form a doubling metric, consider a graph where all nodes have distance 1 to all other nodes.

Observe that it is possible to model a UDG with a UBG by using the Euclidean distances of the UDG and connecting those node pairs which have distance at most 1.

Fact 2.11 *A UDG is a UBG.*

[4] A metric space defines distances between all pairs of nodes, while guaranteeing *non-negativity*, *identity of indiscernibles*, *symmetry* and *triangle inequality*.

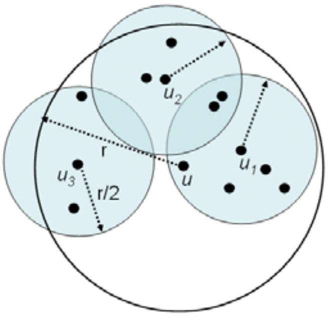

Figure 4. The Euclidean plane forms a doubling metric. In this example, the nodes are distributed in $\mathbb{R}^2$, and three balls of radius $r/2$ are sufficient to cover all nodes in $B_u(r)$, i.e., $B_u(r) = B_{u_1}(r/2) \cup B_{u_2}(r/2) \cup B_{u_3}(r/2)$.

However, as shown below, even the QUDG can be modeled by a UBG.

Fact 2.12 *An undirected QUDG with constant ρ is a UBG.*

Proof. *We transform the distances between all pairs of nodes (u, v) in the QUDG as follows. Let $d_Q(u, v)$ denote the distance from node u to node v in the QUDG, and let $d_B(u, v)$ be the transformed distance in the UBG. Moreover, let $\epsilon > 0$ be an arbitrary small number.*

$$d_B(u,v) := \begin{cases} d_Q(u,v)/\rho & \text{if } d_Q(u,v) \le \rho \\ 1 & \text{if } \rho < d_Q(u,v) \le 1 \text{ and} \\ & v \text{ is adjacent to } u \\ 1+\epsilon & \text{if } \rho < d_Q(u,v) \le 1 \text{ and} \\ & v \text{ is \textit{not} adjacent to } u \\ d_Q(u,v) & \text{if } d_Q(u,v) > 1 \end{cases}$$

Observe that by this transformation, pairs of nodes which are adjacent in the QUDG are assigned distances of at most 1 and are therefore also adjacent in the UBG. Similarly, nodes which are not adjacent in the QUDG have a distance larger than 1 are therefore not neighboring in the UBG either. Also observe that the transformation increases the distance between two nodes by less than a constant factor of $\mu := (1 + \epsilon)/\rho$, but it never decreases any distances. It remains to show that after the transformation, the nodes indeed form a doubling metric space.

In order to form a metric space, the distances between the nodes are to fulfill the following properties: (1) non-negativity, (2) identity of indiscernibles, (3) symmetry and (4) triangle inequality. The non-negativity and the identity of indiscernibles criteria are met trivially. The symmetry criterion however might not hold, as the adjacency relation can be directed in a QUDG. Therefore, in the following, we consider undirected QUDGs only. Hence, since our distance transformation maintains symmetry, Property (3) holds as well. It remains to discuss the triangle inequality.

Consider two arbitrary nodes u and v. Since in the QUDG, all distances are Euclidean, it holds that

$$\forall w : d_Q(u,v) \le d_Q(u,w) + d_Q(w,v) \qquad (1)$$

Let us now look at the following three cases in turn: (i) $d_Q(u,v) \le \rho$, (ii) $\rho < d_Q(u,v) \le 1$, and (iii) $1 < d_Q(u,v)$. In Case (i), no node w with distance larger than ρ from any of the two nodes u and v can challenge the triangle inequality. For all other nodes w, however, it holds that $d_B(u,v) = d_Q(u,v)/\rho \le (d_Q(u,w)+d_Q(w,v))/\rho = d_B(u,w) + d_B(w,v)$. Here, the equalities hold by the definition of the transformation function and the inequality is due to Equation (1). Next, we tackle Case (ii). Again, only nodes w with $d_Q(u,w) \le \rho$ and $d_Q(w,v) \le \rho$ can challenge the inequality. However, we know that $d_Q(u,v) > \rho$, and hence, Equation (1) yields $d_B(u,w) + d_B(w,v) = d_Q(u,w)/\rho + d_Q(w,v)/\rho > 1$. Finally, the triangle inequality also holds in Case (iii), because the distance between u and v in the UBG is the same as in the QUDG, and our transformation never decreases any distances.

We conclude the proof by showing that the metric space has a constant doubling dimension. Recall that all distances are only stretched by a constant factor between 1 and μ in our transformation. Therefore, for all nodes u and arbitrary radii r, $B_u^{QUDG}(r/\mu) \subseteq B_u^{UBG}(r)$. Thus, at most a constant factor of $O(\log \mu)$ times more balls are needed for the UBG than for the QUDG (the Euclidean plane) in the worst case, and the claim follows.

The UBG itself has a polynomially bounded independence and is therefore a BIG.

Fact 2.13 *A UBG is a BIG. [11]*

Proof. *Fix a node u. We have to prove that the total number of independent nodes in $B_u(r)$ grows polynomially in r. Observe that—due to the triangle inequality—in $B_u(1/2)$ there is at most one independent node. Thus, by the definition of a doubling metric, there are at most c independent nodes in $B_u(1)$, at most c^2 in $B_u(2)$, c^3 in $B_u(4)$, etc. Generally, there are at most $c^{\log r+1}$ independent nodes in $B_u(r)$. Since $c^{\log r} \in O(r^c)$, the claim follows.*

To conclude this section, we present two additional modeling aspects with which connectivity models can be extended. The first aspect concerns the nodes' antennas.

Option 2.14 (Antennas) *Besides omnidirectional antennas, there is a wide range of more sophisticated antenna models. For example, a node can have a directional radio antenna with more gain in certain directions.*

Finally, as mentioned in the discussion of the QUDG, links are not always reliable: links may be up and down, e.g., according to a probabilistic process.

Option 2.15 (Link Failures) *Any graph-based model can be enhanced with probabilistic links.*

3 Interference

In wireless networks, the communication medium is shared, and transmissions are exposed to interference. Concretely, a node u may not be able to correctly receive a message of an adjacent node v because there is a concurrent transmission going on nearby. While for higher layer protocols it may be accurate enough to model interference by having random transmission failures, interference must be a first-class citizen for lower layer protocols.

In some sense, an interference model explains how concurrent transmissions block each other. Interference is a difficult phenomenon, with many hard-to-capture characteristics. A signal might for example interfere with itself due to *multipath propagation* (e.g., a direct path canceling with a longer path reflecting on an object). Capturing these effects is beyond the scope of this overview paper, and probably beyond the scope of algorithmic research in general, for some years to come.

Instead models that capture reality from a worst-case perspective have been proposed. The mother of all interference models is the so-called *physical* or *SINR model* [9, 21, 24], which is widely accepted and applied by "lower layer researchers" such as information theorists. In this model, the successful reception of a message depends on the received signal strength, the ambient noise level, and the interference caused by simultaneously transmitting nodes.

Model 3.1 (Signal-to-Interference Plus Noise (SINR)) *Let P_r be the signal power received by a node v_r and let I_r denote the amount of interference generated by other nodes. Finally, let N be the ambient noise power level. Then, a node v_r receives a transmission if and only if $\frac{P_r}{N+I_r} \geq \beta$. Thereby, β is a small constant (depending on the hardware) and denotes the minimum signal to interference ratio that is required for a message to be successfully received. The value of the received signal power P_r is a decreasing function of the distance $d(v_s, v_r)$ between transmitter v_s and receiver v_r. More specifically, the received signal power is modeled as decaying with distance $d(v_s, v_r)$ as $\frac{1}{d(v_s,v_r)^\alpha}$. The so-called* path-loss exponent *α is a constant between 2 and 6 and depends on external conditions of the medium, as well as on the exact sender-receiver distance.*[5] *Let P_i be the transmission power level of node v_i. A message transmitted from a node $v_s \in V$ is successfully received by a node v_r if*

$$\frac{\frac{P_s}{d(v_s,v_r)^\alpha}}{N + \sum_{v_i \in V \setminus \{v_s\}} \frac{P_i}{d(v_i,v_r)^\alpha}} \geq \beta.$$

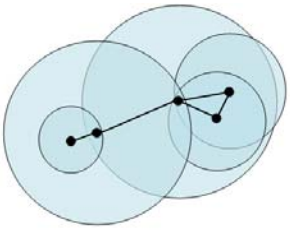

Figure 5. Sample network with heterogeneous transmission ranges. For instance, the node on the left saves energy and minimizes interference by using only a small power level.

Note that sometimes a variation of this SINR model is used in literature. It has an additional requirement: for a successful reception, the received signal power must exceed a minimal threshold θ, i.e., $P_r \geq \theta$. In many situations, such a threshold can also be incorporated implicitly by the ambient noise power level N.

Apart from the interference term, and if all nodes send with the same transmission power level, the connectivity model of SINR is exactly the UDG, with path-loss exponent α and minimum ratio β such that the maximum distance of receiving a signal is 1. Hence, the SINR model can be extended similarly to the UDG model.

Observe that the SINR model does not specify the signal power P_s used by a sender v_s to transmit data to the receiver v_r. Three models are common:

Option 3.2 (Power Control) *CONST: All nodes use the same* constant *transmission power. DIST: The power level depends on the* distance *d between sender and receiver. Concretely, the transmission power is given by $c \cdot d^\alpha$ for some $\alpha \geq 2$ and some constant $c > 0$. GEN: A general (or* arbitrary*) power level is assumed at the sender, which may change over time.*

Figure 5 depicts a network where each node has a different power level.

Although the SINR model incorporates many important physical properties, it has not received an appropriate amount of attention from the algorithms community [21].

[5] In free space, α roughly equals 2. To give another example, in the so-called *two-ray ground model*, it is assumed that there are two paths of the electromagnetic wave: a direct one and a ground reflected signal path; to describe this situation, $\alpha = 4$ is used.

Figure 7. Example where UHI fails: nodes v_1 and v_{k+2} are separated by a path of $k+1$ hops, but are close (distance $1+\epsilon$). Thus, concurrent transmissions of nodes v_2 and v_{k+2} may interfere at v_1 in spite of their large hop distance.

a parameter of the UDG. For instance, some MAC protocols (e.g., coloring algorithms [20]) have been proposed to reduce interference by ensuring a certain hop distance between two senders. Concretely, it is assumed that only the k-neighborhood of a receiver u can interfere with u.

Model 3.5 (UDG with Hop Interference (UHI)) *Nodes are located at arbitrary positions in $\mathbb{R}^2$. Two nodes are adjacent if and only if their Euclidean distance is at most 1. Two nodes can communicate directly if and only if they are adjacent, and if there is no concurrent sender in the k-hop neighborhood of the receiver (in the UDG).*

Clearly, this is a stark simplification since in a UDG a $(k+1)$-neighbor can be close to the receiver, see Figure 7.

Observe that while the UHI model—for every k—sometimes overlooks interference terms which the UDI would take into account, the contrary does not hold.

Fact 3.6 *By choosing $R = k$, and since a hop has at most length 1, the UDI model does not overlook any interference terms which UHI would have taken into account. The contrary does not hold, cf. Figure 7.*

Like UDI and UHI, also the *protocol model* (PM) is of type ONE (Option 3.3). However, the senders in the PM model adapt their transmission power according to DIST (Option 3.2), i.e., depending on the distance between sender and receiver. Model 3.7 is a variation of the model introduced in [9].

Model 3.7 (Protocol Model (PM)) *Let $u_1, u_2, ..., u_k$ be the set of nodes transmitting simultaneously to receivers $v_1, v_2, ..., v_k$ respectively. The transmission of u_i is successfully received by v_i if for all $j \neq i$, it holds that $d(u_j, v_i) > \lambda \cdot d(u_j, v_j)$, where $\lambda \geq 1$ is a given constant. That is, v_i must not fall into a "guard zone" around any sender u_j which is a factor $(1+\lambda)$ larger than u_j's transmission range.*

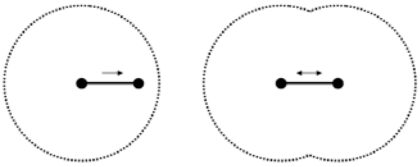

Figure 8. DIR vs. UNDIR: On the left, only the sender transmits data (interference disks around senders). On the right, there is no distinction between sender and receiver, and hence interference arises from the entire link ("pretzels" around links).

Many interference models distinguish between senders and receivers assuming that interference arises at senders and occurs at receivers. However, often receivers acknowledge messages and are therefore also senders. If the original messages are short (e.g., control messages), then the sender/receiver distinction may not make sense. By this observation, some models (e.g. [18]) simply consider the interference of *undirected links*. Figure 8 depicts an example.

Option 3.8 (Direction) *DIR: This class of interference models distinguishes between senders and receivers (interference disks around senders). UNDIR: Interference originates from undirected links (interference "pretzels" around links).*

As in the case of connectivity models, the SINR, the UDI, and the UHI models can be extended with directional antennas and link failures, and hence Options 2.14 and 2.15 also apply here. Moreover, also the idea of quasi unit disk graphs (cf. Model 2.3) could be adopted. For example, the UDI can be "quasified" as follows: if two nodes are closer than a given threshold R_1, concurrent transmissions will always interfere; if the distance is larger than a second threshold R_2, there will be no interference. Finally, if the distance is between R_1 and R_2, transmissions may or may not interfere.

However, these models are often too complicated to be handled algorithmically. It is sometimes simpler to study general weighted graphs instead.

Model 3.9 (General Weighted Graph (GWG)) *Two graphs are given: a weighted connectivity graph G and a weighted interference graph H. For simplicity, it is often assumed that $G = H$. A receiver v successfully receives a message from a sender u, if and only if the received signal strength (the weight of the link between u and v in G) divided by the total interference (the sum or the max of the weights of the links of concurrently transmitting nodes with a receiver v in H) is above the threshold given by the signal-to-interference-plus-noise ratio.*

The general weighted graph model is quite pessimistic, as it allows for non-natural network topologies. Instead—similarly to the BIG connectivity model—we need a weighted graph model that captures the geometric constraints without making too many simplifying assumptions. Again, one approach is to assume that the nodes form a *doubling metric* (cf. UBG model of Section 2).

Model 3.10 (Doubling Metric (DM)) *The DM model assumes that the nodes form a doubling metric, that is, the set of nodes at a distance of at most r from a node u can be covered by a constant number of balls of radius $r/2$ around other nodes for any r (cf. Model 2.9). Interference can be incorporated in various ways. For example, the amount of interference at a receiver u could depend on u's distance (in the doubling metric space) to the closest concurrently transmitting node (ONE model), or on the number of concurrent senders (SUM model).*

Finally note that in this section, we have only studied *binary interference models*, that is, either a message can be received or not. In practice, also the *transfer rate* at which messages can be transmitted can depend on interference: the larger the signal-to-noise ratio, the larger also the available bandwidth.[6] However, since these issues are beyond the scope of this paper, we refer the reader to [6] for more details.

Figure 9 summarizes the connectivity and the interference models.

4 Algorithms

Algorithms for sensor networks come in different flavors. In this section, we first describe a selection of algorithm models which we feel are relevant for sensor networks. We then discuss modeling aspects which may influence an algorithm's performance; for example, what kind of identifiers nodes have, or how the nodes are distributed in space. Besides the classic evaluation criteria for algorithms—namely, *time complexity* and *space complexity*—algorithms for sensor networks pose additional optimization problems; for example, the number of messages which are sent should be small; or, in order to maximize the lifetime of the network, the nodes' energy consumption must be minimized. These issues are also considered in this section.

The first kind of algorithms we present here is similar to the classic (graph) algorithms appearing in the area of theoretical computer science or applied mathematics. These algorithms can operate directly on the entire network or graph, and have complete information about the state of the

[6] A WLAN 802.11 for example exploits environments with less interference in order to transmit more data per time unit.

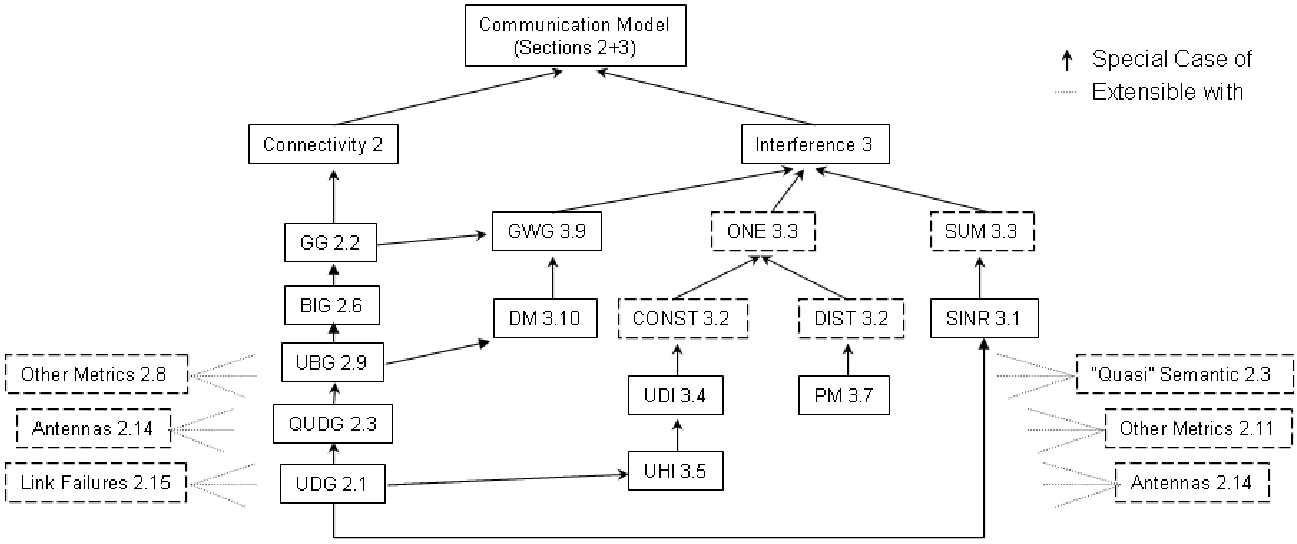

Figure 9. Overview of connectivity and interference models presented in Sections 2 and 3. The arrows show how the models are related. The number behind each model indicates where the description of the model can be found in the text.

system. For example, a system designer planning a fixed sensor network can apply a global algorithm to determine the optimal positions of the nodes in a given observation area.

Model 4.1 (Global Algorithms) *A* global algorithm *can operate directly on the entire network.*

However, most algorithms for sensor networks proposed in literature are meant to be *executed by the sensor nodes* during the system's operation. For example, when a node receives a message, it performs some computation, and—depending on the computation's results—sends a new message to one of its neighbors. A node a priori only knows its own state. In order to learn more about the other nodes in the network, it has to communicate with its neighbors. By collaboration of the nodes, global operations such as (multi-hop) routing between two nodes can be achieved. Since the activity is distributed among the nodes, these algorithms are called *distributed algorithms* [22].

Model 4.2 (Distributed Algorithms) *In a* distributed algorithm, *every sensor node runs its own algorithm. A priori, a node has only information about its own state. In order to learn more about the rest of the network, nodes repeatedly exchange messages with adjacent nodes.*

Distributed algorithms raise many interesting questions. For example: What can be computed in a distributed fashion, and what not? How efficient is a distributed algorithm compared to a corresponding global algorithm? Besides an algorithm's correctness, execution time to perform the task (*time complexity)* and memory requirements at the nodes (*space complexity*), a new criterion becomes important, namely *message complexity*: Since distributed algorithms rely on message passing, and since sending and receiving messages is an expensive operation (e.g., queueing delay, congestion, energy consumption, etc.), a distributed algorithm should minimize the total number of messages sent.

The notion of message complexity motivates the introduction of *localized algorithms* [27].

Model 4.3 (Localized Algorithms) *A* localized algorithm *is a special case of a distributed algorithm. At the beginning, a node has only information about its own state. In order to learn more about the rest of the network, messages have to be exchanged. In a k-*localized algorithm, *for some constant k, each node is allowed to communicate at most k times with its neighbors. However, a node can decide to retard its right to communicate; for example, a node can wait to send messages until all its neighbors having larger identifiers have reached a certain state of their execution.*

In spite of the more restricted communication model, localized algorithms can be slow. A node u might have to wait for a neighbor v to transmit all its messages, while node v in turn has to wait for its neighbor w, etc. Thus, as a matter of fact there can be a *linear chain of causality*, with only one node being active at any time. This yields a worst-case execution time of $\Theta(n)$, where n is the number of nodes.

Researchers have proposed to study yet another kind of distributed algorithms which overcome the performance problems of localized algorithms and always terminate after a constant number of communication rounds [15].

Model 4.4 (Local Algorithms) *Again, at the beginning, each node only knows its own state. In a k-*local algorithm*, for some constant k, each node can communicate at most k times with its neighbors. However, in contrast to k-localized algorithms, nodes cannot delay their decisions. In particular, all nodes process k synchronized phases, and a node's operations in phase i may only depend on the information received during phases* 1 *to* $i-1$.

Observe that in a k-local algorithm, nodes can only gather information about nodes in their k-neighborhood. The constant time complexity makes Model 4.4 particularly suited in scenarios where the nodes' environment changes frequently. However, due to the synchronous phases, local algorithms may make greater demands on the MAC layer than localized algorithms.

Having defined the most common types of algorithms, we now look at some algorithmic aspects in more detail. As mentioned, the message complexity—the total number of messages sent by an algorithm—is a main evaluation criterion of distributed algorithms. As the number of messages typically depends on the amount of information which can be stored in a message, a model must specify the messages' sizes. A most popular model limits the message size to $O(\log n)$ bits, where n is the total number of nodes in the system. Hence, a message can store only a constant number of node identifiers (e.g., the source and destination address of a routing packet). Moreover, it is often assumed that if a node u sends a message to a neighbor v, all other neighbors of u will also receive the message (*broadcast model*). However, sometimes—for example for lower bound proofs [12]—also models are considered where the message size is *unbounded*, and where nodes can communicate with their neighbors individually (*message-passing model*).

Algorithmic models also differ in their assumptions about how nodes can access the wireless medium. The concrete MAC however can influence the number of retransmissions and hence an algorithm's performance. Moreover, sometimes an algorithm must be able to coordinate the medium access itself.

Option 4.5 (Medium Access) *Some researchers assume an ideal medium access mechanism [26] where interference is impossible, and where messages will always be broadcast instantaneously to all neighbors (cf. models of Section 2). In addition, adversarial models are used where an adversary schedules transmissions. Of course, this model only makes sense if the adversary is restricted appropriately, that is, if there are fairness guarantees. For example, the adversary might have to schedule each node at least once every $\Theta(n)$ rounds. Moreover, one could imagine an adversary which delivers a message only to a subset of a node's neighbors, because the other neighbors experience collisions. Finally, completely unstructured radio networks [19] can be considered where the algorithm designer has to implement her own medium access scheme from scratch. These models can further be classified in whether collisions can be detected by a receiver or not.*

A main objective of sensor networks is to collect physical data distributed over a given region. To achieve this, typically one or more nodes observe different sub-areas. Knowledge of the nodes' distribution however can be important for an algorithm designer. In a scenario where the nodes are dropped from an airplane, one might expect that the nodes are roughly randomly distributed when they reach ground.

Model 4.6 (Random Node Distribution) *The simplest—and quite common—way to model sensor networks is to assume a UDG in combination with a* uniform node distribution *in the 2-dimensional Euclidean plane. However, motivated by percolation theory, also* Poisson models *have been proposed [5]: Thereby, the positions of the nodes are distributed in $\mathbb{R}^2$ according to a homogeneous Poisson point process of constant density λ per unit area.*

While these random models may be fine to prove the performance of an algorithm, for correctness and robustness issues, a more pessimistic model should be preferred, e.g., a worst-case distribution.

Model 4.7 (Worst-Case Node Distribution) *Nodes are distributed arbitrarily in the space given by the underlying graph (e.g., Euclidean plane, general graph, etc.).*

Of course, there are again many models which lie between the two extremes. For example, random distributions with a density parameter varying over space could be considered: One can imagine that there are several nodes per square meter in areas which are "interesting" to observe, whereas in other "routing only" areas nodes are hundreds of meters apart. Finally, in the context of node distribution, one has to mention also models which do not allow nodes to be arbitrarily close or even assume the same position; there is such an assumption in the $\Omega(1)$ *model* [17] or in so-called *civilized graphs* [10].

Related to the distribution of nodes in space is also the issue of the distribution of node *identifiers*. As many algorithms are based on node IDs, their performance can depend on how IDs are distributed among the nodes (and thus also in space).

Options 4.8 (Node Identifiers) *We propose that nodes* *have unique identifiers.*[7] *Similarly to the node distribution in space, the most common models for ID distributions are* random distributions *and* worst case distributions. *Sometimes, it also matters from which range the identifiers are chosen. Again, many variations are possible. For example, each of the* n *nodes can have a unique 128-bit identifier (range* $0, ..., 2^{128} - 1$*). Or, in a more restrictive case, the nodes may have consecutive numbers (e.g., range* $1, ..., n$*).*

Alternatively,—or additionally!—node IDs can contain location information, for example if the nodes are equipped with a *Global Positioning System* (GPS) or a *Galileo* device. Location information can boost the performance of certain operations [17]: for example, a routing algorithm can exploit geographic information to forward the message to a neighbor which lies in the direction of the message's destination (greedy routing).

Option 4.9 (Location Information) *Sensor nodes can have access to various forms of (absolute or relative) geographic information about other nodes. For example, a node* u *might sense its distance to another node* v*, or in which direction (angle of arrival)* u *lies, or even know* v*'s exact position.*

Distributed algorithms for sensor networks are usually evaluated according to their time complexity, their space complexity and their message complexity. However, in order to be successful in a real sensor network, an algorithm has to pursue additional objectives. For example, if sensor nodes are deployed in large numbers, recharging their batteries seems out of question, in particular in adversarial territory. A node's energy supply must suffice for the whole operational phase. Therefore, the conservation of energy is of utmost importance. Basically, there are two approaches to capture the energy consumption of a node. Historically,—since during the transmission of data much energy is consumed—a model has been studied which only takes the transmission energy into account [4].

Model 4.10 (Transmission Energy) *The energy consumed by a node is calculated by the sum over all its transmissions. Thereby, the energy needed to transmit one message is of the form* $c \cdot d^{\alpha}$*, where* d *is the distance between sender and receiver,* α *is the path-loss exponent (usually* $\alpha > 2$*), and* c *is a constant.*

Although transmitting data is a costly operation, sensor nodes with short-range radios available today spend as much energy receiving or waiting for data. Therefore, techniques have been developed which allow nodes to change to a parsimonious *sleep mode* [23]. During the time periods a node is sleeping, it cannot receive any data. The idea is that if all nodes can somehow be synchronized to wake up at the same moment of time to exchange data (e.g., every minute), much energy is saved. This motivates the following model.

Model 4.11 (Sleeping Time) *The energy consumed by a node is given by the accumulated time in which it is not in sleep mode.*

If there are no external disturbances, a node is assumed to live as long as it has some energy left. The lifetime of the entire network is modeled in different ways.

Option 4.12 (Network Lifetime) *In applications which depend on every single node, the lifetime of a network can be defined as the time until the first node runs out of battery power [2]. Alternatively, a network might be able to tolerate certain node failures; for example, the network might live as long as all live nodes are still connected to each other.*

Finally, note that several variations discussed in the section about interference also apply for energy models. For example, a node might be able to save energy by using a smaller transmission radius if the receiver is close-by. Last but not least, interference itself causes a waste of energy as retransmissions are required, and may hence be incorporated into a given energy model.

5 Conclusion

This paper has presented and compared a subjective selection of algorithmic models. We want to emphasize that there is no optimal model, and that an engineer has to choose the model which reflects her needs best. For example, a large warehouse has different physical characteristics and signal propagation paths than an office building; or GPS might not work indoors and hence algorithms based on coordinates are not be feasible; etc. We hope this paper helps to compare the different options. Generally, we believe that for efficiency considerations, a slightly idealistic model can be fine. However, when it comes to issues such as correctness of an algorithm, it seems that a more pessimistic or conservative model should be preferred. In other words, a *robust* algorithm is also correct in a more general model than for which it has been studied or proven efficient, i.e., in a model which is located closer to the "root" in Figure 9 (according to the transitive "→" relation).

[7] Some researchers also study scenarios where there are no node IDs. However, observe that IDs could be generated during deployment, for example by a random number generator. Moreover, as already RFID tags have IDs, we believe it is reasonable to assume that sensor nodes obtain a unique ID already during the production process. Finally, note that certain tasks cannot be solved by any distributed algorithm if there are no identifiers, as there is no way to break symmetries among the nodes.

Acknowledgments

The authors would like to thank Fabian Kuhn, Thomas Moscibroda, and Pascal von Rickenbach for many valuable discussions.

References

[1] L. Barrière, P. Fraigniaud, and L. Narayanan. Robust Position-based Routing in Wireless Ad Hoc Networks with Unstable Transmission Ranges. In *Proc. 5th Intl. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM)*, pages 19–27, 2001.

[2] J.-H. Chang and L. Tassiulas. Energy Conserving Routing in Wireless Ad-hoc Networks. In *Proc. IEEE Infocom*, pages 22–31, 2000.

[3] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit Disk Graphs. *Discrete Mathematics*, 86:165–177, 1990.

[4] A. E. F. Clementi, G. Huiban, and P. Penna. On the Approximation Ratio of the MST-Based Heuristic for the Energy-Efficient Broadcast Problem in Static Ad-Hoc Radio Networks. In *Proc. 17th International Symposium on Parallel and Distributed Processing (IPDPS)*, Washington, DC, USA, 2003.

[5] O. Dousse, P. Thiran, and M. Hasler. Connectivity in Ad-hoc and Hybrid Networks. In *Proc. IEEE Infocom*, 2002.

[6] A. El Fawal, J.-Y. Le Boudec, R. Merz, B. Radunovic, J. Widmer, and G. M. Maggio. Trade-off Analysis of PHY-Aware MAC in Low-Rate Low-Power UWB Networks. *IEEE Communications Magazine*, December 2005.

[7] D. Ganesan, D. Estrin, A. Woo, D. Culler, B. Krishnamachari, and S. Wicker. Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks. Technical report, UCLA Computer Science Technical Report UCLA/CSD-TR 02-0013, 2002.

[8] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded Geometries, Fractals, and Low-Distortion Embeddings. In *Proc. 44th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*, page 534, 2003.

[9] P. Gupta and P. Kumar. The Capacity of Wireless Networks. *Trans. Information Theory*, 46(2):388–404, 2000.

[10] S. O. Krumke, M. V. Marathe, and S. S. Ravi. Models and Approximation Algorithms for Channel Assignment in Radio Networks. *Wireless Networks*, 7(6):575–584, 2001.

[11] F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer. Fast Deterministic Distributed Maximal Independent Set Computation on Growth-Bounded Graphs. In *Proc. 19th Intl. Symp. on Distributed Computing (DISC)*, 2005.

[12] F. Kuhn, T. Moscibroda, and R. Wattenhofer. What Cannot Be Computed Locally! In *Proc. 23rd ACM Symp. on the Principles of Distributed Computing (PODC),*, 2004.

[13] F. Kuhn, T. Moscibroda, and R. Wattenhofer. On the Locality of Bounded Growth. In *Proc. 24th ACM Symp. on Principles of Distributed Computing (PODC*, pages 60–68, 2005.

[14] F. Kuhn, T. Nieberg, T. Moscibroda, and R. Wattenhofer. Local Approximation Schemes for Ad-hoc and Sensor Networks. In *Proc. Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, pages 97–103, 2005.

[15] F. Kuhn and R. Wattenhofer. Constant-time Distributed Dominating Set Approximation. In *Proc. 22nd Annual Symp. on Principles of Distributed Computing (PODC)*, pages 25–32, 2003.

[16] F. Kuhn, R. Wattenhofer, and A. Zollinger. Ad-hoc Networks Beyond Unit Disk Graphs. In *Proc. 1st ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, 2003.

[17] F. Kuhn, R. Wattenhofer, and A. Zollinger. Worst-Case Optimal and Average-case Efficient Geometric Ad-hoc Routing. In *Proc. 4th ACM Int. Symp. on Mobile Ad Hoc Networking & Computing (MobiHoc)*, pages 267–278, 2003.

[18] F. Meyer auf de Heide, C. Schindelhauer, K. Volbert, and M. Grünewald. Energy, Congestion and Dilation in Radio Networks. In *Proc. 14th Annual ACM Symp. on Parallel Algorithms and Architectures (SPAA)*, pages 230–237, 2002.

[19] T. Moscibroda, P. von Rickenbach, and R. Wattenhofer. Analyzing the Energy-Latency Trade-off during the Deployment of Sensor Networks. In *Proc. IEEE Infocom*, 2006.

[20] T. Moscibroda and R. Wattenhofer. Coloring Unstructured Radio Networks. In *Proc. 17th Annual ACM Symp. on Parallelism in Algorithms and Architectures (SPAA)*, pages 39–48, 2005.

[21] T. Moscibroda and R. Wattenhofer. The Complexity of Connectivity in Wireless Networks. In *Proc. IEEE Infocom*, 2006.

[22] D. Peleg. *Distributed Computing: A Locality-sensitive Approach*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[23] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava. Energy Aware Wireless Microsensor networks. *IEEE Signal Processing Magazine*, 19(2):40–50, 2002.

[24] T. Rappaport. Wireless Communications: Principles and Practices. *Prentice Hall*, 1996.

[25] I. Stojmenović, A. Nayak, and J. Kuruvila. Design Guidelines for Routing Protcols in Ad Hoc and Sensor Networks with a Realistic Physical Layer. *IEEE Communications Magazine*, March 2005.

[26] P.-J. Wan, K. M. Alzoubi, and O. Frieder. Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks. *Mobile Networks and Applications*, 9(2):141–149, 2004.

[27] Y. Wang, X.-Y. Li, P.-J. Wan, and O. Frieder. Sparse Power Efficient Topology for Wireless Networks. *J. of Parallel and Distributed Computing*, 2002.