On Communication Models for Algorithm Design in Networked Sensor Systems: A Case Study ⋆,⋆⋆

Yang Yu, Bo Hong, Viktor K. Prasanna

Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089-2562

Abstract

Towards building a systematic methodology of algorithm design for applications of networked sensor systems, we formally define two link-wise communication models, the Collision Free Model (CFM) and the Collision Aware Model (CAM). While CFM provides ease of programming and analysis for high level application functionality, CAM enables more accurate performance analysis and hence more efficient algorithms through cross-layer optimization, at the expense of increased programming and analysis complexity. These communication models are part of an abstract network model, above which algorithm design and performance optimization is performed. We use the example of optimizing a probability based broadcasting scheme under CAM to illustrate algorithm optimization based on the defined models. Specifically, we present an analytical framework that facilitates an accurate modeling and analysis for the probability based broadcasting in CAM (PB_CAM). Our analytical results indicate that (1) the optimal broadcast probability for either maximizing the reachability within a given latency constraint or minimizing the latency for a given reachability constraint decreases rapidly with node density, and (2) the optimal probability for either maximizing the reachability with a given energy constraint or minimizing the energy cost for a given reachability constraint varies slowly between 0 and 0.1 over the entire range of the variations in node density. Our analysis is also confirmed by extensive simulation results.

Key words: Networked sensor systems, Algorithm design, Communication model, Probability based broadcasting

1 Introduction

Networked sensor systems(NSSs) are ad-hoc self-organizing networks of tiny sensors characterized by severe energy constraints. State-of-the-art wireless communication in NSSs exhibits two important features, including much higher energy cost relative to computation and packet collision between concurrent communication due to signal interference. These two features make energy aware algorithm design with carefully managed communication strategy a crucial and challenging component of large-scale NSS applications [1, 2].

Contemporary approaches for algorithm design and optimization of NSS applications are centered around manual customization of the network protocol stack. However, such approaches are simply not feasible for future large-scale NSSs primarily because the large number of protocols executing concurrently in NSSs – positioning, topology maintenance, medium access control, time synchronization, calibration, error detection, routing, and the application-level functionality – make it very difficult to optimize the design while simultaneously ensuring correct operation. Therefore, systematic methodologies for designing algorithms for NSS applications are needed to cope with the continuing advancements in sensor node design and increasingly complex applications.

Specifically, a systematic algorithm design methodology will eventually enable domain experts to design, analyze, and optimize (manually or automatically) algorithms based on an abstract network model of the underlying system, without requiring a knowledge of lower level networking and hardware aspects of the system. Figure 1(a) illustrates a general view of the expected design methodology centered around an abstract network model [3]. Such a network model, essentially a layer of abstraction between what we term 'low-level' details of the network and 'high-level' concerns of the algorithm designer, includes information on network deployment, communication model, basic computation and communication primitives supported by the model, and a set of cost functions associated with the primitives. Algorithm design based on the abstract network model offers advantages including ease-of-design, rapid performance analysis, enhanced portability, and system level optimization, at the cost of possible performance loss due to the layering overhead.

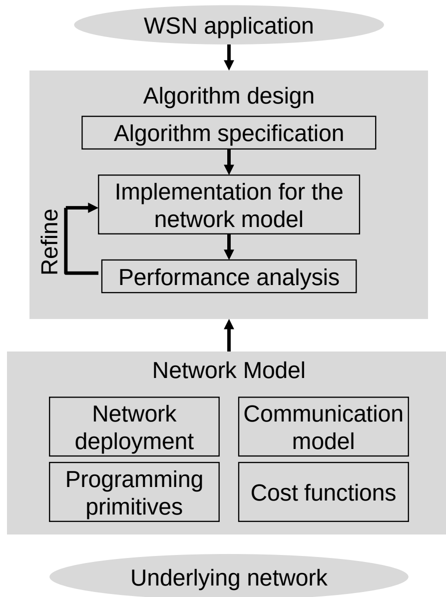
The proposed design methodology is as follows. First, the network model for

---

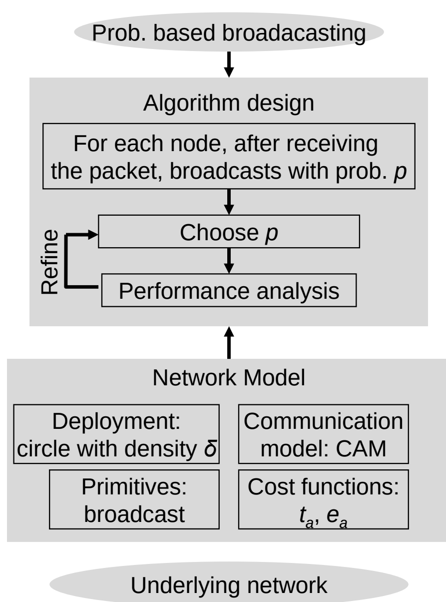
⋆ This work is supported by NSF under grant IIS-0330445.

⋆⋆A preliminary version of this paper has been accepted by the Workshop of Advances in Parallel and Distributed Computational Models (APDCM) 2005. This version contains additional materials including analytical and simulation results for energy cost.

Email address: {yangyu, bohong, prasanna}@usc.edu (Yang Yu, Bo Hong, Viktor K. Prasanna).



(a) General design methodology



(b) A case study: optimizing a probability based broadcasting scheme

Fig. 1. Algorithm design methodology for NSS applications

the target class of NSSs is defined using a bottom-up approach that involves analysis of the expected nature of deployment, characteristics of the node hardware, etc. The end user specifies an algorithm according to the desired functionality and the defined network model. The algorithm is then optimized using a top-down approach as follows. The algorithm is specified using an architecture-independent application model such as an annotated task graph. The algorithm specification is used as input to an optimization tool that will (manually or automatically) *explore alternate design time and run time adjustment of algorithmic parameters and determine an efficient one based on the network model and user-specified performance metrics*. Such an optimization procedure requires certain mechanisms to model the behavior of the specified algorithm based on the network model and to analyze and optimize its performance accordingly.

In this paper, as an initiative step towards our goal of building a systematic algorithm design methodology for NSS applications, we define two simple models for abstracting link-wise communication in NSSs and use the example of optimizing a probability based broadcasting scheme based on the defined models to illustrate the concept of algorithmic performance modeling and optimization. The success of PRAM family in distributed and parallel systems [4] as well as the model based algorithm design in reconfigurable computation [5] motivate us to develop communication and computation models in a similar spirit. This paper serves the purpose of conveying the importance of developing such a systematic algorithm design methodology and illustrating its feasibility from the perspectives of system modeling and algorithm optimization.

In the following two subsections, we summarize the content of this paper.

1.1 The Definition of Two Communication Models

We formally define two models that explore the tradeoffs between ease of programming and the power of providing accurate performance analysis. These models are not new in themselves as they have been widely used in the literature. Our contribution is a formal definition of these models with a comparison of their pros and cons for algorithm design in NSSs.

Our first model, the Collision Free Model (CFM) abstracts reliable communication as an atomic operation with all details of lower level channel behavior being transparent to the algorithm designers. Our second model, the Collision Aware Model (CAM) specifies that packet collision occurs when multiple senders try to communicate to their common neighbors at the same time. The motivation behind these two models is that while CFM is designed for applications that are concerned more about ease of programming and high-level functionality, CAM is for applications that need to cope with lower level network behaviors for cross-layer optimization.

Specifically, CFM is powerful in terms of programming, which essentially bears resemblance to algorithm design in traditional parallel and distributed computation. However, CFM does not really capture the impact of packet collision that distinguishes wireless communication from wired communication, which makes performance analysis under CFM inaccurate or even misleading.

On the other hand, compared with CFM, CAM models current technologies in wireless communication (e.g., broadcast in 802.11) more closely. Consequently, CAM shifts all responsibility for detecting and handling packet collisions to the algorithm designer, which requires extra programming efforts to ensure reliable packet delivery. However, the exposure of certain lower level details also offers opportunities to perform more accurate performance analysis, in terms of both functionality and energy-efficiency, and hence to design more efficient algorithms for real-life networks.

1.2 A Case Study: Optimizing Probability Based Broadcasting in CAM

Broadcasting distributes a piece of information from a source node to all nodes in the network, which is a fundamental operation for many existing algorithms and protocols in NSSs [6–8]. We consider two simple broadcasting schemes that are suitable for NSSs, simple flooding and probability based broadcast. Although programming and analysis of simple flooding in CFM is easy, the information provided by CFM is not sufficient for a realistic and accurate performance analysis.

On the contrary, the exposure of low level information on packet collisions in CAM enables a realistic and accurate performance analysis for the broadcasting schemes. Since simple flooding is a special case of probability based broadcasting, we are only concerned about probability based broadcasting. We instantiate the components in Figure 1(a) and depict in Figure 1(b), the optimization of the probability based broadcasting scheme. Specifically, the network deployment is abstracted as a set of sensor nodes uniformly distributed in a circular field with node density δ; the communication model is CAM; the programming primitives is broadcast; and the cost functions is the time and energy costs of each broadcast in CAM. The specification for the probability based broadcasting is straightforward. By treating the broadcast probability, p as a tunable algorithmic parameter, the optimization of the algorithm essentially requires to choose a proper p based on the performance analysis according to user-defined performance metrics and the network model.

To do so, we propose an analytical framework to determine the optimal broadcast probability with respect to 4 performance metrics that involve reachability, and latency, and energy cost. Specifically, our analytical results indicate that (1) the optimal broadcast probability for either maximizing the reachability within a given latency constraint or minimizing the latency for a given reachability constraint decreases rapidly with node density, and (2) the optimal probability for either maximizing the reachability with a given energy constraint or minimizing the energy cost for a given reachability constraint varies slowly between 0 and 0.1 over the entire range of the variations in node density. More importantly, by choosing an appropriate broadcast probability, PB_CAM scales very well with node density. These results are important to the design and implementation of time and energy efficient probability based broadcasting in practice. We also show that this results matches very well to our simulation results.

Note that although the models defined in this paper are essentially at link layer and consequently the primitives available for algorithm design is broadcast or unicast, the proposed methodology for algorithm design is open to models at higher levels, which may also enrich the body of primitives. For example, primitives for reliable point-to-point communication or group communication under transport layer models are expected to be investigated in our future work. However, to implement such primitives in the corresponding models and identify the proper cost functions for such primitives are challenging and beyond the scope of this paper.

1.3 Paper Organization

We briefly discuss related work in Section 2. Our system assumptions and formal description of CFM and CAM are presented in Section 3. In Section 4, we describe our analytical framework to model and optimize the probability based broadcasting under CAM. The simulation results are demonstrated in Section 5. Finally, we give concluding remarks in Section 6.

2 Related work

Although a large amount of literature exists about algorithms to perform various tasks on NSSs, little work has been dedicated to defining well-defined models of the underlying communication or computation mechanisms. Nevertheless, most existing works for algorithm design in NSSs either explicitly or implicitly assume their own link-wise communication models with different sets of assumptions. Hence, the assumptions and models described in this paper have been widely used in various contexts. However, the lack of formal definition and systematic composition of these assumptions and model specifications remains an obstacle for either fair comparison between different works or the development of systematic design methodologies based on unified and commonly accepted computation and communication models.

Due to its ease of programming and analysis, CFM is adopted by numerous existing works that focus on high level algorithm design or application development, including in-network processing and data gathering [9–13], performance analysis [14,15], localization and time synchronization algorithms [16,17], and other topics [18, 19]. Most of the above works focus on high level functionality or semantics of the studied problems. Hence, the hypothesis here is that in real systems with packet collisions, the designed algorithms can achieve a performance that is close to its analytical prediction in CFM.

The packet collision model in CAM is also widely considered in the literature. One approach to cope with packet collision is to ensure collision free communication at algorithmic level. Such works include data routing in a single-hop network [20,21], cooperative data distribution in multi-hop networks [22,23], or capacity analysis [24]. When the impact of collision is considered, similar models to CAM are used for modeling system-wide performance [25], or characterizing the behavior of broadcasting with directional antennas [26].

The models presented in this paper are from a relatively high level perspective that facilitates efficient algorithm design in NSSs. Hence, low level details about hardware devices, protocol implementation, and physical layer signal propagation are abstracted away. Many existing works have addressed such low level details [27,28]. For example, models that deal with hardware devices such as MICA nodes from Berkeley can be found in [28]. Also, a physical layer channel access model for large dense packet radio networks is described by Shepard [27].

Since broadcasting is a basic operation in large scale networks, the behavior of broadcasting has been previously studied [29–31]. A nice categorization of broadcasting schemes were presented by Williams *et. al.* [30], which consists of simple flooding, probability based scheme, area based scheme, and neighbor knowledge scheme. Due to the simplicity in their implementation and analysis, we focus on the simple flooding and probability based schemes in this paper. Nevertheless, the extension of our analytical framework for the area based scheme and neighbor knowledge scheme is part of our future work.

While an analytical study of the reachability of simple flooding is given in [31], the performance of the probability based scheme has only been studied through simulation [29, 32]. Simulations in [29] show that when latency constraint is not considered, the broadcast probability has little impact on the reachability when the network is dense (around 100 neighbors per node on the average) and high broadcast probability is required to achieve a high reachability when the network is sparse (for less than 10 neighbors per node on the average). Also, results in [29] show that the latency of broadcasting increases rapidly with the broadcast probability. By applying percolation theory, it is indicated in [32] that for a grid deployment of sensor network with collision free communication, the optimal broadcast probability should be around 0.59. Moreover, for general networks with random sensor node deployment and non-collision free communication, simulation results in [32] indicate that for a fixed node density, the reachability shows a bell curve when the broadcast probability increases, with the maxima reached by lower broadcast probability when the node density increases.

In our study, we propose 4 performance metrics that are important in the context of NSSs and present an analytical framework to reveal the optimal probability for optimizing each of the 4 performance metrics. To the best of our knowledge, our work is the first one that tries to analytically model and analyze the performance of the probability based broadcasting in general wireless networks with random node deployment and a collision aware communication model. For the performance metric of reachability under latency constraint, our analytical results conform the simulation results in [32]. Also, for the performance metric of latency under reachability constraint, our analytical results conform the simulation results in [29].

3 Our Models

3.1 System Assumptions

We make the following assumptions about the underlying network.

(1) The system consists of a set of n homogeneous sensor nodes, V. All sensor nodes in V have identical transmission radius r. The same time cost is required for sending or receiving a packet of unit size. Also, the same energy cost is required for sending or relieving a packet of unit size.
(2) The deployment of the network is represented as a symmetric graph $G(V, E)$, where E is the set of edges (or communication links). Edge $(u, v) \in E$ means that u and v are within the communication range of each other. Hence, u and v are referred to as neighbors of each other.
(3) All nodes have locally unique IDs and every node knows the IDs of all its neighbors.
(4) We assume the availability of certain power management schemes by which radios on every node are switched off when it does not need to participate in any communication. Hence, the energy cost of the algorithms considered in this paper only accounts for the cost of sending and receiving packets.
(5) We consider a stable snapshot of the system, where the mobility or hard-

ware failure of nodes are not modeled.

(6) To model packet collisions in CAM, we assume a simple collision model where all concurrent packet transmissions to the same destination collide with each other. In other words, a packet transmission is successful only if it is the only packet transmission to the receiver throughout the whole transmission time duration.

Assumption 1 implies that when interference is not considered, the signal to noise ratio (SNR) remains high up to a certain distance r from the sender, enabling nearly perfect reception of the transmitted signal. However, SNR drops rapidly beyond this distance r, resulting in unacceptable bit error rates at the receiver. Although this assumption does not incorporate the fluctuation in SNR due to shadowing and multi-path fading, it provides a clear high-level abstraction that facilitates algorithm design at application level. The assumption of same energy cost for sending and receiving a packet is reasonable for short range communication in highly dense networks, where the power consumption is dominated by the radio electronics [33]. The symmetric communication assumed in Assumption 2 also serves the purpose of abstracting low level details for high level algorithm design. There assumptions are widely used in papers listed in Section 2.

Assumption 3 is widely used for distributed algorithm design and protocol implementation in NSSs [8]. A stronger assumption is that each sensor node has a globally unique ID. Such an assumption might be needed for complex applications that require network-wide collaboration. Nevertheless, the assumption of locally unique ID suffices to illustrate the example of broadcasting in this paper.

Assumption 4 basically eliminates the energy cost of sensor nodes in idle state. Radios on sensor nodes are switched on only if they need to communicate with other nodes. This is easy to realize at the senders, but not the receivers since the receivers do not have the knowledge in advance of possible communication in the future. Hence, certain mechanisms including ultra-low power paging channel are needed [34]. The time and energy cost of these mechanisms is assumed to be absorbed into the cost of communication for data packets.

For our purpose, we do not consider system dynamics such as node mobility and node or link failure in our model, as stated by assumption 5. In fact, such dynamics can be captured by the changes in the topology of the network graph, which is beyond the scope of this paper.

Assumption 6 is also frequently used by many works that focus on algorithm design to model packet collision (as described in Section 2). Although in real-life communication scenarios, packet reception is usually determined by the SNR ratio at the receiver, our assumption hides such MAC layer details from the algorithm designers. This model also facilitates our statistic analysis of the probability of a successful packet transmission with respect to an application level packet transmission strategy. In the literature, the concept of carrier sensing range [35] (which is typically twice of the transmission range) is also used to state that concurrent communication within the carrier sensing range also leads to packet collision. Although our analysis assumes packet collision in transmission range, it can be extended to incorporate the concept of carrier sensing range (please refer to Appendix A for details). We also believe that similar analytical and simulation results will be obtained for collision models based on carrier sensing range, since it does not change the essential part of the collision model — more concurrent communication leads to higher probability of packet collision.

Note that we do not assume time synchronization for the network. Algorithm design under this assumption is challenging, since communication among nodes may happen in an asynchronous fashion. However, while algorithms should be designed so that they work properly in the worst case of asynchronous behavior, we analyze the performance of the algorithms from an optimistic perspective where perfect time synchronization is assumed.

3.2 Model Description

We consider two basic types of communication primitives – broadcast and unicast. Since our models can be applied to both primitives, we do not distinguish between them in the following descriptions.

3.2.1 Collision Free Model (CFM)

In CFM, each packet transmission is modeled as an atomic operation that is guaranteed to succeed with time cost t_f and energy cost e_f. Based on assumption 1 in Section 3.1, the same time and energy costs apply to both the sender and the receivers.

CFM is powerful in high level algorithm design and programming since it allows fully parallelized packet transmission. By ignoring all low-level details of contention resolution, the incentives in the model lead the designer to expose the maximum possible computational and communication parallelism of a given task. However, the above time and energy parameters are too simple to properly reflect the cost of low level contention resolution, which is tightly related to the deployment of the network and the implementation of MAC layer. This shortcoming restricts the ability of CFM to provide an accurate performance analysis.

CFM can be implemented in several ways. A naive implementation based on CSMA/CA style protocols (such as 802.11) is to require acknowledgment from all receivers of each broadcasting and re-transmit the packet if timeout occurs. This implementation usually leads to significant network traffic for acknowledging a broadcast, and hence high time and energy costs. Other implementations include the use of multi-packet reception (MPR) techniques through time, code, and frequency diversity. For example, TDMA exploits the time diversity by assigning to each sensor node a specific time slot that is ideally unique in its neighborhood. Each sensor node can communicate only in the assigned time slot. However, such MPR techniques require additional hardware and more complicated coordination among sensor nodes, which might not be affordable for large scale networks with tiny sensor nodes. Hence, most existing algorithms in NSSs are still based on CSMA/CA protocols, which we will also focus on in this paper.

3.2.2 Collision Aware Model (CAM)

In CAM, packet transmissions performed by low level communication protocols are not guaranteed to succeed. Specifically, when a sensor node is the target for concurrent communication operations (including both broadcast and unicast) from multiple neighbors, none of the communication operations succeeds. Thus, the information about packet collisions are exposed to algorithm designers who are now also free to choose the way of handling contentions in network media access. Let t_a and e_a denote the time and energy costs of a packet transmission. In general, we have $t_a \leq t_f$ and $e_a \leq e_f$.

There are two ways to handle packet collision. If reliable communication is required by the application, it is the responsibility of application level algorithms to provide either collision avoidance or collision detection and retransmission mechanisms. However, such mechanisms inevitably lead to high energy cost due to the contention-based media access. On the other hand, there also exist certain applications that can gain advantage of the high redundancy in the network such that the effect of packet loss is negligible or the applications can tolerate certain degree of packet loss and still function properly solely based on the packets that are successfully delivered. From this perspective, broadcasting is a nice example that we will study in next section.

CAM describes the exact behavior of broadcast and unicast of small size packets in 802.11, where packet transmission is performed without RTS/CTS/ACK mechanisms.

4 A Case Study: Optimizing Probability Based Broadcasting in NSSs

The broadcasting problem considers the distribution of a piece of information from a source node to the whole network. For instance, the source node can be the base station, where user queries are injected into the network.

To focus on a general and yet realistic network deployment as well as to make the analysis tractable, we consider a uniform deployment of N nodes in a circle of radius Pr, where r is the communication range of nodes and P is a pre-specified integer parameter. We also assume that the source node is placed at the center of the circle. Let δ denote the density of the network, i.e., the average number of nodes in unit area. We have $N = \delta\pi(Pr)^2$. Since the scalability of various broadcasting schemes is of particular importance from algorithm design perspective, we vary δ during the optimization of the broadcasting algorithm.

We study two simple schemes from the literature [29]. The first scheme is the *simple flooding* scheme, where each node broadcasts exactly once after it receives the information from any neighbor for the first time. The second scheme is the *probability based broadcasting* scheme, where after receiving the broadcasted packet, each node broadcasts exactly once with probability p and does not broadcast with probability $1 - p$.

Simple flooding fits perfectly into CFM where packet collision is abstracted away. For the considered network with N sensor nodes within a circle of radius Pr, it is easy to see that simple flooding achieves a reachability (defined below in Section 4.1) of 1 with time cost $O(Pt_f)$ and energy cost $O(Ne_f)$. While simple flooding is easy to program and analyze in CFM, various studies have revealed that the above performance analysis is inaccurate in real-life systems where packet collisions cannot be ignored [29]. This is mainly because the cost functions t_f and e_f rely heavily on node density; but unfortunately, their inter-relationship is not effectively abstracted in CFM.

On the other hand, the exposed low level information of packet collisions in CAM enables a realistic and accurate performance analysis for the broadcast schemes of interest. In this section we present an analytical framework that serves the above purpose, which is the key component of automatic performance optimization in Figure 1(b). Before proceeding with our analytical framework, we first define a set of performance metrics that are of particular interest in NSSs.

4.1 Performance Metrics

Two performance metrics have been previously studied: *reachability* and *latency* [29–31]. Specifically, reachability is defined as the fraction of sensor nodes that receive the information when the broadcasting algorithm terminates, while the corresponding time cost is defined as the latency.

Due to the severe energy limitation in NSSs, we propose a set of performance metrics that take reachability, latency, as well as energy cost into consideration. Basically, given any one of these three quantities as a constraint, we can evaluate a broadcasting scheme based on its capability of optimizing the rest two quantities. Hence, we have the following 6 performance metrics:

(1) the reachability given a latency constraint,
(2) the energy cost given a latency constraint,
(3) the latency given a reachability constraint,
(4) the energy cost given a reachability constraint,
(5) the reachability given an energy constraint, and
(6) the latency given an energy constraint.

To minimize the second performance metric or to minimize the last metric is not meaningful, since both of them lead to the trivial solution that no broadcast should be performed. Hence, we use the rest 4 metrics to evaluate a broadcasting scheme. Intuitively, to maximize the reachability given a latency constraint (the first metric) is a dual of the problem of minimizing th latency given a reachability constraint (the third metric); to minimize the energy cost given a reachability constraint (the fourth metric) is a dual of the problem of maximizing the reachability given an energy constraint (the fifth metric). While these 4 metrics are trivial to optimize for simple flooding in CFM, no formal analysis has been performed for the probability based broadcasting in CAM to optimize these 4 metrics.

For our study, we use the number of broadcasts, denoted as M as an indirect metric to measure the energy cost. This is because the total energy cost can be calculated by multiplying M with e_f and also the number of expected number of neighbors of each node, which is $\delta \pi r^2$. By measuring M instead of the total energy, we essentially allow the total energy budget to scale with δ, which is reasonable for a fair comparison.

4.2 Modeling and Optimizing Probability Based Broadcasting in CAM

To make our analytical framework applicable to more general scenarios, we also consider a simple backoff method that can be easily incorporated into our broadcasting scheme. Specifically, the algorithm is executed in consecutive time phases, with each phase consisting of s time slots. We assume that the length of each time slot is sufficiently long for performing a broadcast. After receiving the information from a neighbor node, each node chooses a random timeslot during the next time phase, in which the information is broadcasted with probability p. This method models the commonly used jitter technique [30] that intentionally delays the broadcast at each node by a random time duration to reduce packet collision. We refer to the above probability bases scheme as PB_CAM.

Note that PB_CAM does not require synchronized time slots and time phases at various nodes. The algorithm begins when the source node sends out the packet. However, solely for the purpose of analysis, we assume strict time synchronization across the network so that the time slots at all nodes are perfectly aligned. The basic idea of our analysis is to partition the entire sensor field into P concentric rings of width r. We track the execution of the algorithm and estimate the expected number of sensor nodes in each ring during each time phase. In the following text, we first show some preliminaries that are useful in modeling PB_CAM.

4.2.1 Preliminaries

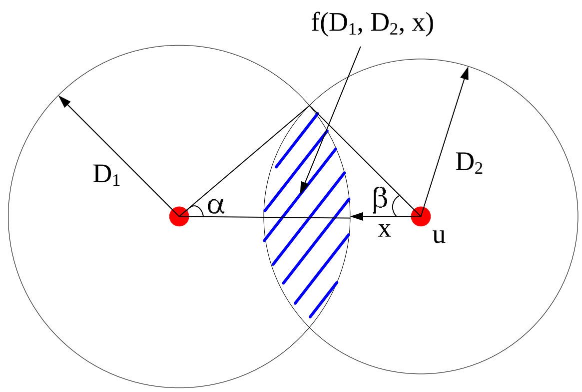


Fig. 2. The intersection of two circles

Consider two intersecting circles L_1 with radius D_1 and L_2 with radius D_2 (Figure 2). Let x denote the distance between the center of L_2, u and the border of L_1 (x is positive if u is outside L_1 and negative if u is inside L_1). Let $f(D_1, D_2, x)$ denote the area of intersection. We have:

$$\alpha = \arccos\left(\frac{D_1^2 + (D_1 + x)^2 - D_2^2}{2D_1(D_1 + x)}\right), \text{ and } \beta = \arccos\left(\frac{D_2^2 + (D_1 + x)^2 - D_1^2}{2D_2(D_1 + x)}\right).$$

Hence, the area of intersection, $f(D_1, D_2, x)$ can be calculated as:

$$f(D_1, D_2, x) = \alpha D_1^2 - D_1^2 \sin\alpha\cos\alpha + \beta D_2^2 - D_2^2 \sin\beta\cos\beta \ . \qquad (1)$$

The above calculation will be used to estimate the number of sources that may contend to broadcast to a common receive in the same time phase.

We now study the case where K nodes need to send a packet to a common destination in a time phase of s time slots and analyze the probability for the destination to successfully receive at least one packet from any of the K senders. Mathematically speaking, consider the problem of randomly dropping $K > 0$ identical items into $s > 0$ identical buckets. We are interested in the probability of having at least one bucket to hold exactly one item, denoted by $\mu(K, s)$. Such a probability does can be solved using the following recursive representation:

$$\mu(K,s) = \begin{cases} 1 & \text{if } K = 1, \quad (2a) \\ K(\frac{(s-1)^{K-1}}{s^K}) + (\frac{s-1}{s})^K \mu(K, s-1) + & \\ \quad \sum_{i=2}^{K-1} \binom{K}{i} \frac{(s-1)^{K-i}}{s^K} \mu(K-i, s-1) & \text{otherwise.} \quad (2b) \end{cases}$$

The rationale behind the above equation is that if we have exactly one item (i.e., $K = 1$), the desired probability is always one; otherwise, by considering the problem based on how many items are dropped into the first bucket, the desired probability can be calculated as the sum of three terms, with each term corresponding to one of the following cases:

- Exactly one item is dropped in the first bucket. In this case, all possible dropping of other $K-1$ items into the remaining $s-1$ buckets are considered to be valid. The probability of such a case is $K(\frac{(s-1)^{K-1}}{s^K})$.
- No item is dropped in the first bucket. The probability of such a case is $(\frac{s-1}{s})^K$, under which, we consider the subproblem of dropping all K items into the remaining $s - 1$ buckets.
- i items are dropped in the first bucket, where $i = 2, \ldots, K - 1$. For a given i, the probability of having i items dropping into the first bucket is $\binom{K}{i} \frac{(s-1)^{K-i}}{s^K}$, under which, we consider the subproblem of dropping $K - i$ items into $s - 1$ buckets. The total probability is then summed up over all possible values of i.

We are not aware of a closed form solution to the above recursion. Hence, the value of $\mu(K, s)$'s are numerically calculated based on the recursion.

4.2.2 Analysis

As previously stated, we regarded the network as a composition of P concentric rings of equal width r. We number the rings as $R_1, R_2, \ldots, R_P$ from the center. Let C_i denote the area of ring R_i, i.e., $C_i = \pi r^2(i^2 - (i-1)^2)$.

We assume strictly aligned time phases, with $T_1, T_2, \ldots$ denote the time phases from time 0 when the source node broadcasts to its neighbors. Let n_j^i denote the expected number of nodes that are located in ring R_j and receive the broadcasted information during T_i. Since the source node is the only node that broadcasts in T_1, all nodes in ring R_1 can successfully receive the information in T_1. Hence, we have $n_1^1 = \delta\pi r^2$, and $n_j^1 = 0$ for $j = 2, 3, \ldots, P$.

Now consider a node u in ring R_j with a distance of $x \in [0, r]$ from the inner boundary of R_j. Let $A(x,k)$ denote the area in ring R_k that is within distance r from u, i.e., all nodes in $A(x,k)$ can reach u by broadcasting. For u in ring R_j, it is clear that $A(x,k)$ is not empty only for $k = j-1, j, j+1$. We have:

$$\begin{aligned}
A(x, j-1) &= f(r(j-1), r, x) \\
A(x, j) &= f(rj, r, x-r) - A(x, j-1) \\
A(x, j+1) &= \pi r^2 - A(x, j-1) - A(x, j)
\end{aligned}$$

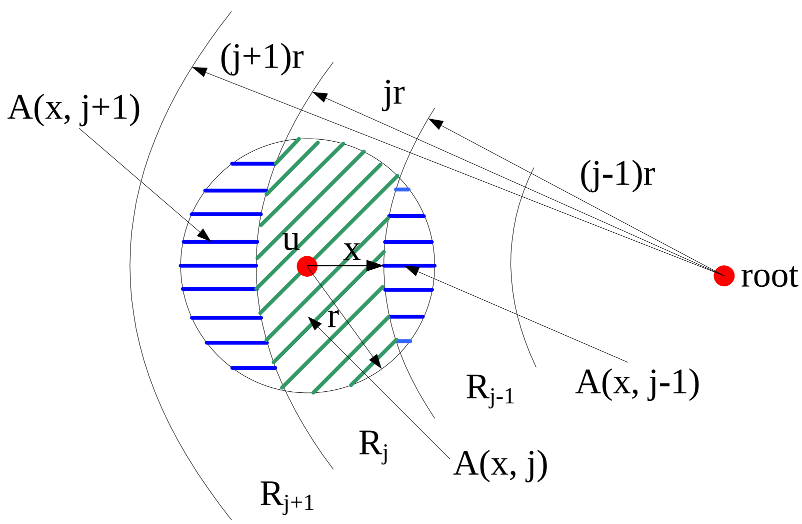


Fig. 3. The partition of the communication range of u

These three portions of area actually form a partition of the area covered by u's communication radius (Figure 3). Now suppose that u has not received the broadcasted information after T_{i-1}. The number of nodes in rings R_{j-1}, R_j, and R_{j+1} that have received the information in T_{i-1} is given by n_{j-1}^{i-1},

n_j^{i-1}, and n_{j+1}^{i-1}. For the sake of our analysis, we assume that these nodes are uniformly distributed in the three rings. We can calculate the expected number of nodes in the communication range of u that have received the broadcasted information in T_{i-1}, denoted as $g(x)$. We have

$$g(x) = \sum_{k=j-1}^{j+1} (n_k^{i-1} \frac{A(x,k)}{C_k}) \ . \tag{3}$$

Consider a small rectangular area $dx \times dy$ whose center is at distance x from the inner boundary of R_j. We have assumed that the nodes in ring R_j that have received the information after T_{i-1} are uniformly distributed in R_j, which implies that the nodes in R_j that have not received the information are uniformly distributed too. The total number of nodes in the rectangular area that have not received the information is therefore $\frac{\delta C_j - \sum_{k=1}^{i-1} n_j^k}{C_j} dxdy$ (recall that C_j is the area of R_j and $\sum_{k=1}^{i-1} n_j^k$ is the total number of nodes in R_j that have received the information till T_{i-1}). For each of such nodes in area $dx \times dy$, its probability of successfully receive at least one packet is $\mu(g(x)p, s)$ (recall that p is the probability for each node to broadcast). By integrating $\mu(g(x)p, s)\frac{\delta C_j - \sum_{k=1}^{i-1} n_j^k}{C_j} dxdy$ over the area covered by R_j, we can obtain the value for n_j^i. Therefore, we have the following result (with slight modification to suit polar coordinates since R_j is circular).

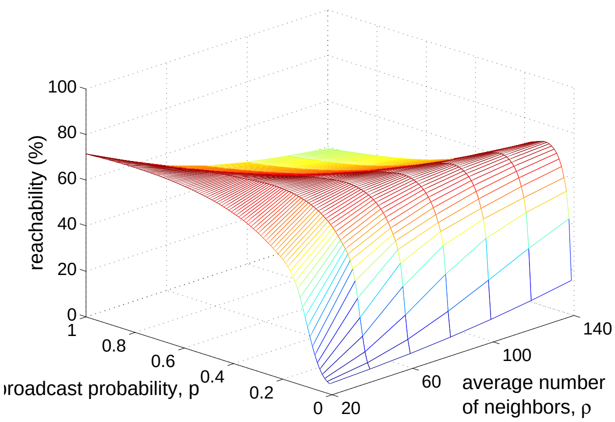
$$n_j^i = \int_0^{2\pi} \int_0^{r} (r(j-1)+x)\mu(g(x)p, s) \frac{\delta C_j - \sum_{k=1}^{i-1} n_j^k}{C_j} dx d\theta \ . \tag{4}$$

Equation (4) does not have a closed form solution. Hence, for a given i, the above representation of n_j^i needs to be recursively applied over $j = 1, 2, \ldots, P$ to calculate the number of nodes in each ring that have received the information by the end of the i-th time phase. Therefore, given a network configuration in terms of P, s, and δ, we are now able to model the behavior of PB_CAM for the optimization of the 4 performance metrics described in Section 4.1.

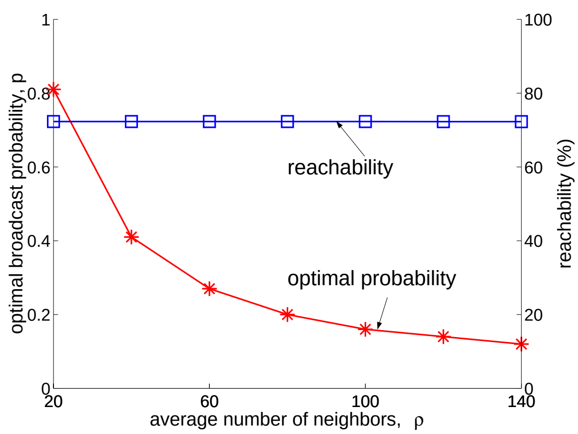
4.2.3 Maximize the Reachability for a Given Latency Constraint

To incorporate the effect of the communication radius r, we represent the node density using the average number of neighbors within the communication range of a node, denoted as ρ. That is, $\rho = \delta \pi r^2$ (ignoring the boundary effect). We will use the terms "node density" and "average number of neighbors" exchangeably hereafter. We consider a network with $P = 5$ and vary ρ from

20 to 140 in increments of 20. Hence, the total number of sensor nodes in the entire network scales from around 500 up to 4000. We also assume $s = 3$ and set p from 0.01 to 1 in increments of 0.01. In Figure 4(a), we show the reachability of PB_CAM within a latency constraint of 5 time phases. It can be observed that the for various value of ρ, the maximal reachability is achieved at certain probability. Also, for a fixed ρ, the reachability shows a bell curve when p increases, which conform the simulation results in [32].



(a) Reachability with respect to variations in ρ and p



(b) Optimal probability with the corresponding reachability

Fig. 4. Reachability of PB_CAM in 5 time phases

In Figure 4(b) we plot the optimal probability that maximizes the reachability within 5 time phases as a function of ρ, with the corresponding reachability. It clearly shows that the optimal probability is a decreasing function of ρ. While the curve of the optimal probability drops fast when ρ is small, it becomes flat when ρ is large. Surprisingly, we also observe that the corresponding reachability is consistently around 72%. This result indicates that the impact of node density to reachability within a given latency constraint can be diminished by choosing a proper broadcast probability. As a comparison, the curve with probability 1 in Figure 4(a) is actually the reachability of simple flooding in CAM, which is around 0.55 of the optimal when $\rho = 140$.

4.2.4 Minimize the Latency for a Given Reachability Constraint

We consider the same network configuration as the one described in the previous section. We assume that a 72% reachability constraint is required by the users. Also, we assume that the number of nodes that receive the information in each time phase is evenly distributed across the time dimension within the time phase. Hence, we are able to get a continuous measurement of latency using fractions of time phases.

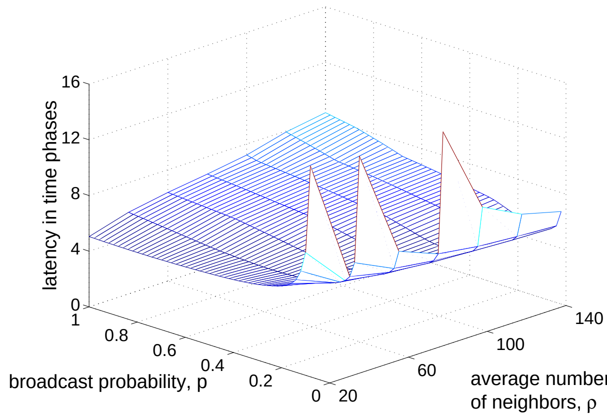
In Figure 5, we show the latency in time phases for achieving 72% reachability. When the probability p is close to 0, we found that for some combinations of p and ρ, it is impossible to achieve a reachability of 72% because too few nodes are expected to broadcast in each time phase. Hence, the latency for such combinations is not shown.

The optimal probability for minimizing the latency to achieve 72% reachability is shown in Figure 5(b). We observe that regardless of the variations in ρ, 72% reachability can be achieved in 5 time phases by carefully adjusting the broadcast probability. However, from Figure 5(a), simple flooding (the curve with $p = 1$) uses more than 8 time phases when $\rho = 140$. Also, the optimal probability curve in Figure 5(b) is the same as the one in Figure 4(b). This is understandable, since to maximize the reachability given a latency constraint is a dual of the problem of minimizing the latency given a reachability constraint.

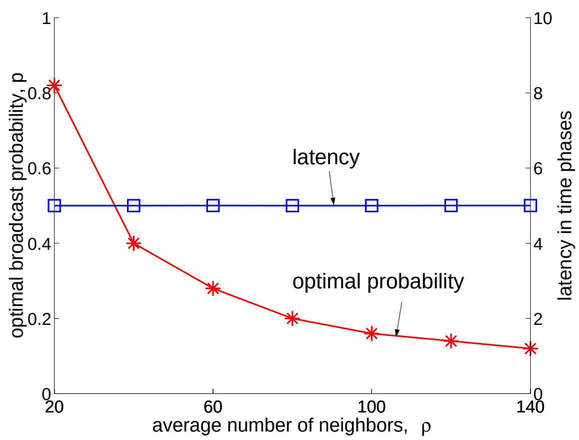
4.2.5 Minimize the Energy for a Given Reachability Constraint

As previously stated, we use the number of broadcasts to measure the energy cost. For a reachability of 72%, the number of broadcasts with respect to variations in ρ and p is illustrated in Figure 6(a). For combinations of ρ and p that cannot achieve a 72% reachability, the number of broadcasts is not shown. From Figure 6(a) we observe that the number of broadcasts predominantly increases with both ρ and p, which is quite understandable.

We are more interested in the probabilities that minimize the energy cost, which is plotted in Figure 6(b). Surprisingly, the optimal probability varies slowly between 0 and 0.1 over the entire range of the variations in node density,



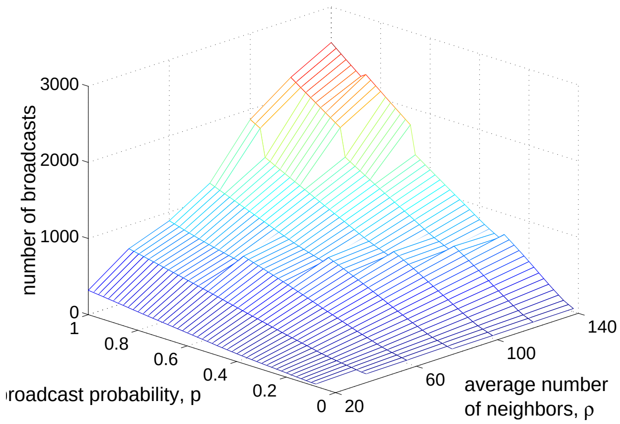
(a) Latency with respect to variations in ρ and p



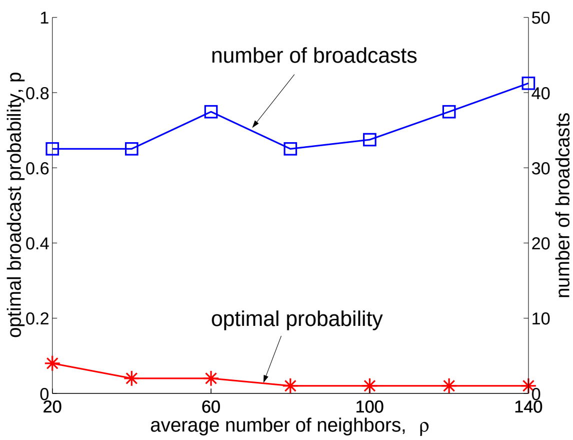
(b) Optimal probability with corresponding latency

Fig. 5. Latency of PB_CAM for achieving 72% reachability

thus implying its insensitivity to node density. This is quite different from the curve of the optimal probability in Figure 5(b). The reason is that to minimize energy costs essentially requires the minimization of packet collision. Since no latency constraint is imposed, low broadcast probability is favored as long as the required reachability can be achieved. In fact, we do observe latencies ranging from 7 to 15 time phases resulting from the optimal probability. Also, the optimal broadcast number is within 40 throughout variations in ρ, which is almost one percent of the 3000 broadcasts for simple flooding with $\rho = 1.4$.



(a) Number of broadcasts with respect to variations in ρ and p

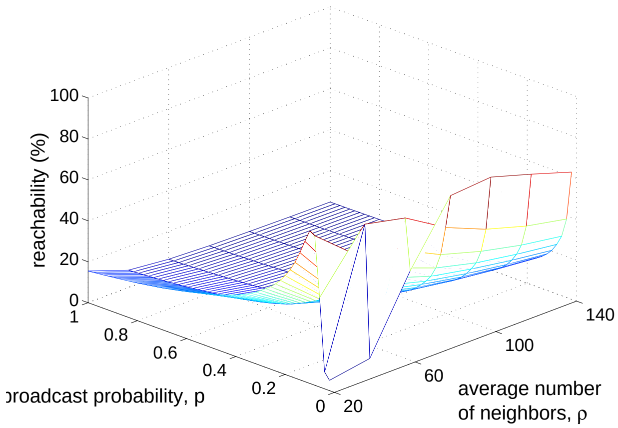


(b) Optimal probability with the corresponding number of broadcasts

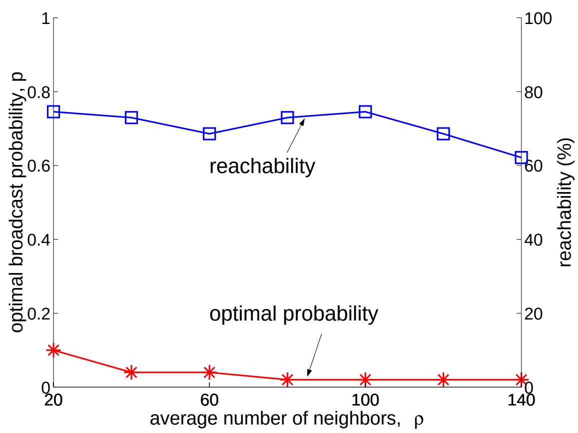
Fig. 6. Energy cost of PB_CAM for achieving 72% reachability

4.2.6 Maximize the Reachability for a Given Energy Constraint

We set the number of allowed broadcasts to 35 and analyze the reachability with respect to variations in ρ and p. The results are shown in Figure 7. It can be observed from Figure 7(a) that the maximal reachability is reached when p is close to 0. We further plot the optimal probability in Figure 7(b). We observe that this probability is very close to the optimal probability given in Figure 6(b). This is because to maximize the reachability given an energy constraint is a dual of the problem of minimizing the energy given a reachability constraint. Also, the maximal reachability is around 70%; whereas the reachability for simple flooding is less than 20% (Figure 7(a)).



(a) Reachability with respect to variations in ρ and p



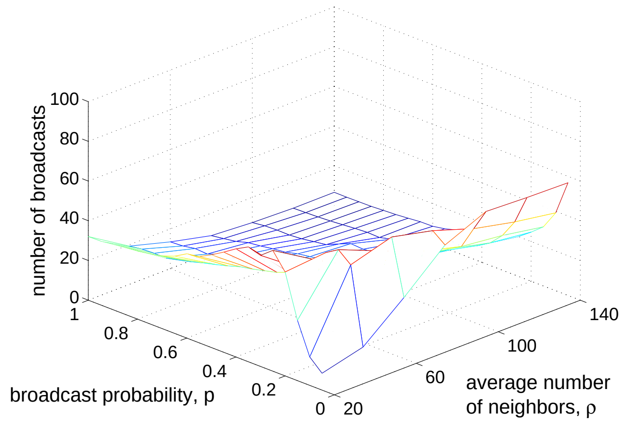
(b) Optimal probability with the corresponding reachability

Fig. 7. Reachability of PB_CAM using ≤ 35 broadcasts

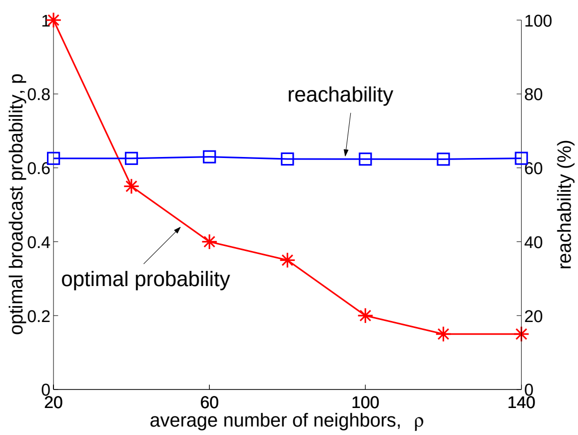
5 Simulation Results

To validate the results obtained in Section 4.2, we performed extensive simulations using the framework provided by GloMoSim [36]. The parameter settings for our simulation were the same as those used for the analysis in Section 4.2, except that for a reasonable running time, we varied broadcast probability from 0.05 to 1 in increments of 0.05. The data shown in this section is averaged over 30 random runs.

We first illustrate the reachability for 5 time phases in Figure 8. It can be observed that our simulation results match the analytical results in Figure 4 quite well. The optimal probability decreases with a similar trend as the curve



(a) Reachability



(b) Optimal probability and corresponding reachability

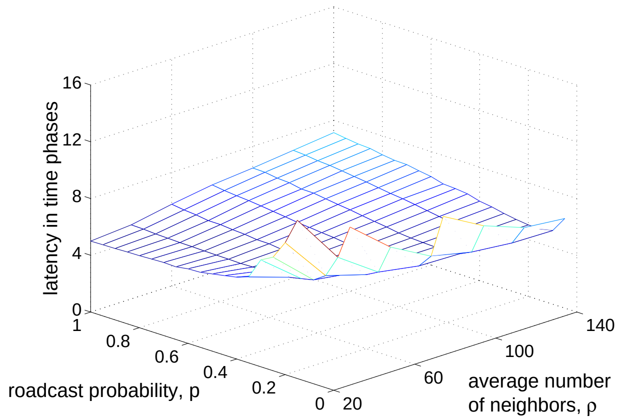
Fig. 8. Simulation results of the reachability of PB_CAM in 5 time phases

in Figure 4(b). Also, the achievable reachability is consistently around 63% with respect to variations in ρ.

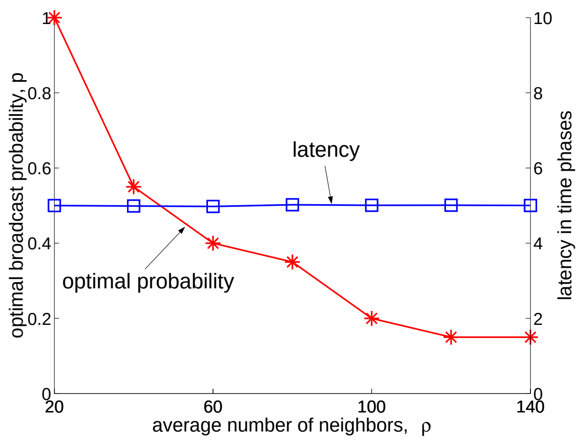
In Figure 9, we show the latency needed to achieve a reachability of 63%. It can be observed that the optimal broadcast probability is very close to the optimal probability in Figure 8(b) and the corresponding latency is 5 time phases. This also confirms our analytical results in Figure 5.

The energy cost in terms of the number of broadcasts required to achieve a reachability of 63% is shown in Figure 10. It can be observed that the optimal probability is within 0.2 throughout the variations in ρ and the corresponding number of broadcasts are around 80.

Finally, in Figure 11, we show the reachability achieved with no more than 80



(a) Latency with respect to variations in ρ and p



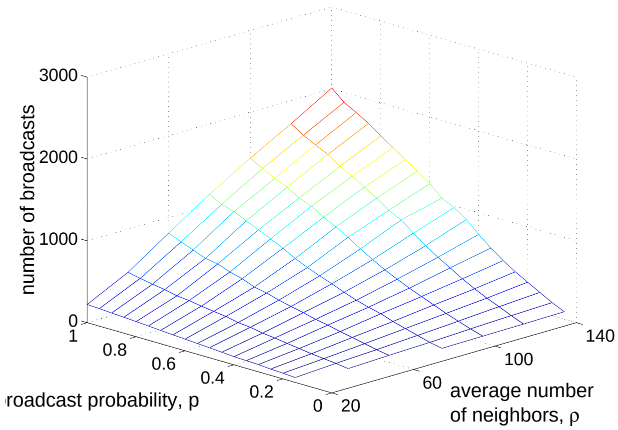
(b) Optimal probability and corresponding latency

Fig. 9. Simulation results of the latency of PB_CAM for 63% reachability

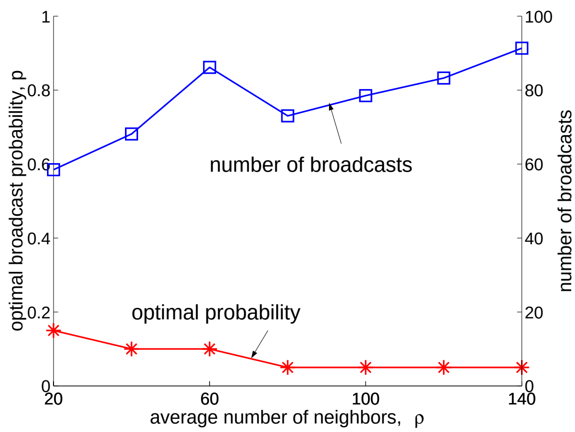
broadcasts. It can be observed that the optimal probability is almost within 0.2 throughout the variations in ρ. In short, our simulation results in Figures 10 and 11 confirm our analytical results presented in Section 4.2.

6 Concluding Remarks

Working towards the goal of building a systematic methodology for algorithm design in networked sensor systems, we have examined two link-wise communication models, the Collision Free Model (CFM) and the Collision Aware Model (CAM). While CFM facilitates high-level algorithm design and perfor-



(a) Number of broadcasts with respect to variations in ρ and p

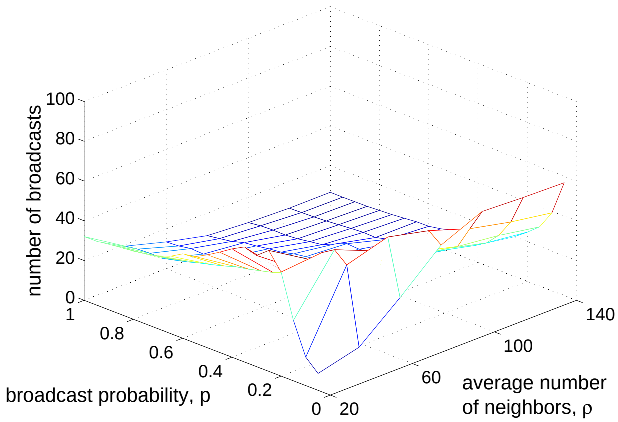


(b) Optimal probability and corresponding number of broadcasts

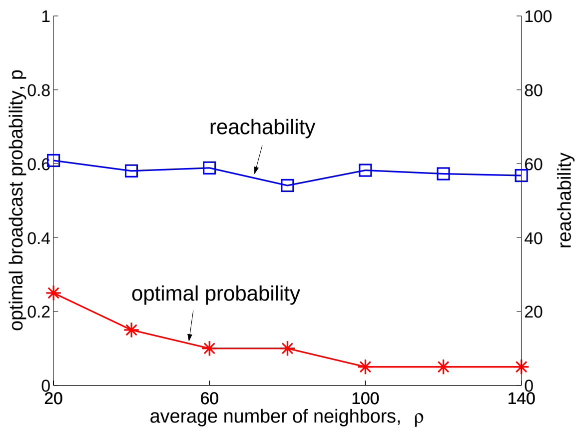
Fig. 10. Simulation results of the energy cost of PB_CAM for 63% reachability

mance analysis, its over-simplification on packet collision restricts its capability to provide an accurate performance prediction in real-life systems. On the other hand, by exposing details of packet collision in CAM, more accurate behavioral modeling and performance analysis of the target applications are possible, which also enables cross-layer performance optimization at the cost of increased programming and analysis complexity.

As a case study, we have described an analytical framework for modeling, analyzing, and optimizing the performance of a probability based broadcasting scheme in CAM (PB_CAM). Specifically, our analytical results indicate two important facts. The first fact is that the optimal probability for PB_CAM to either maximize reachability within a latency constraint or minimize latency



(a) Reachability with respect to variations in ρ and p



(b) Optimal probability and corresponding number of broadcasts

Fig. 11. Simulation results of reachability of PB_CAM using ≤ 80 broadcasts

to satisfy a reachability constraint decreases rapidly with respect to node density; whereas the optimal probability to maximize reachability within an energy constraint or minimize energy cost with a reachability constraint varies slowly between 0 to 0.1 throughout the variations in node density. The second fact is that by carefully choosing the broadcast probability, PB_CAM exhibits very good scalability for optimizing the above 4 performance metrics.

However, the above accurate performance modeling and analysis based on CAM is gained at the expense of disclosing extra information on low-level communication to the algorithm designer and exporting to the algorithm designer the responsibility of handling packets collisions, which increases the complexity of algorithm design. We believe that tradeoffs between CFM and CAM can be explored by defining models that provide reasonable capabilities of cross-layer performance analysis and optimization without laying too much burden on algorithm designers. For instance, by modeling the time/energy costs of a successful packet transmission in CFM as a function of the node density to account for necessary re-transmission serves the purpose of incorporating the impact of network density without getting details of packet collision in the underlying network. To identify proper cost functions is part of our future work. A similar method is to model the success rate of each packet transmission as a function of node density. Such a rate can be used to either derive the expected cost of a successful packet transmission, or specify the corresponding behavior of applications if packet loss is tolerable.

It is also possible to further expose low level information in order to design and implement more efficient algorithms. For the example of broadcasting, we consider the following interesting phenomena. For simple flooding in CAM, we define the success rate of a broadcast as the ratio of the number of nodes that successfully receive the broadcast over the total number of neighbors of the sender. Using our analytical framework, we calculate the average success rate of the simple flooding in CAM. In Figure 12, we compare this rate against the optimal probability depicted in Figure 4(b).

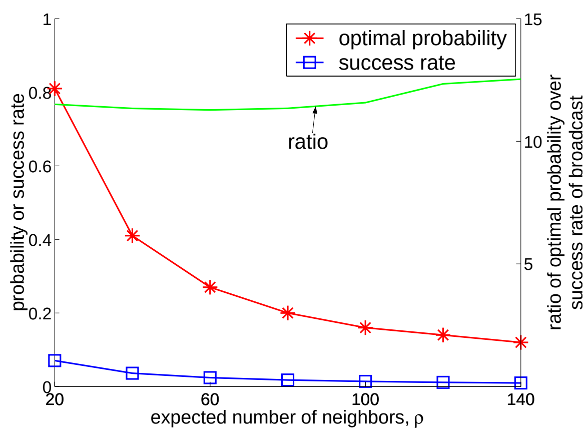


Fig. 12. Comparison of the broadcasting success rate in simple flooding and the optimal probability in Figure 4(a)

It can be observed that the ratio of optimal probability to the success rate is very close to 11 throughout the variations in node density. This strong correlation indicates the possibility of determining the optimal broadcast probability based on the success rate, without the knowledge of node density. This discovery is practically useful if the node density exhibits large spatio-temporal variation. Although practical implementations and costs to obtain the success rate is still an open problem, it is interesting to explore the opportunities to expose low level information in communication models for directing and/or optimizing upper level algorithm design.

A natural research direction that complements algorithm design is the synthesis of designed algorithms into executables of NSSs that specify node-level behavior and system-level cooperation. The concept of design automation that enables automatic synthesis of programs is first proposed by Bakshi *et. al.* [3], which is also centered around a abstract network model. Hence, further research on network and system models are crucial for the development of both algorithm design and automatic synthesis of NSS applications.

Acknowledgment

Many helpful discussions and comments on the earlier draft of the paper by Amol Bakshi, Animesh Pathak, Mitali Singh are gratefully acknowledged. We also thank the anonymous reviewers for their very helpful comments and suggestions.

References

[1] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, K. Pister, System architecture directions for networked sensors, in: 9th International Conference on Architectural Support for Programming Languages and Operating Systems, 2000.

[2] M. Singh, V. K. Prasanna, System level energy tradeoffs for collaborative computation in wireless networks, in: IEEE IMPACCT Workshop, 2002.

[3] A. Bakshi, V. K. Prasanna, Algorithm design and synthesis for wireless sensor networks, in: 33rd International Conference on Parallel Processing, 2004.

[4] M. Maggs, L. R. Matheson, R. E. Tarjan, Models of parallel computation: A survey and synthesis, in: 28th Hawaii International Conference on System Sciences (HICSS), Vol. 2, 1995, pp. 61–70.

[5] R. Vaidyanathan, J. L. Trahan, Dynamic Reconfiguration: Architectures and Algorithms, Kluwer Academic, 2003.

[6] D. B. Johnson, D. A. Maltz, Mobile Computing, Kluwer Academic Publishers, 1996, Ch. Dyanmic Source Routing in Ad Hoc Wireless Networks, pp. 153–181.

[7] C. Perkins, E. M. Royer, Ad hoc on-demand distance vector routing, in: 6th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), 1999, pp. 90–100.

[8] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed Diffusion: A scalable and robust communication paradigm for sensor networks, in: ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), 2000.

[9] B. Hong, V. K. Prasanna, Optimizing a class of in-network processing applications in networked sensor systems, in: 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), 2004.

[10] W. Choi, S. K. Das, Trade-off between coverage and data reporting latency for energy-conserving data gathering in wireless sensor networks, in: International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 2004, pp. 503–512.

[11] W. Choi, S. K. Das, A framework for energy-saving data gathering using two-phase clustering in wireless sensor networks, in: ACM MobiQuitous Networking Conference, 2004, pp. 203–212.

[12] Y. Yu, B. Krishnamachari, V. K. Prasanna, Energy-latency tradeoffs for data gathering in wireless sensor networks, in: IEEE InfoCom, 2004.

[13] W. Choi, S. K. Das, A novel framework for energy-conserving data gathering in wireless sensor networks, in: IEEE InfoCom, 2005.

[14] Z. Cheng, W. Heinzelman, Flooding strategy for target discovery in wireless networks, in: International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems, 2003, pp. 33–41.

[15] S. Pattem, B. Krishnamachari, R. Govindan, The impact of spatial correlation on routing with compression in wireless sensor networks, in: ACM/IEEE International Symposium on Information Processing in Sensor Networks, 2004.

[16] N. Patwari, A. O. H. III, M. Perkins, N. S. Correal, R. J. O'Dea, Relative location estimation in wireless sensor networks, IEEE Transactions on Signal Processing, Special Issue on Signal Processing in Networks 51 (8) (2003) 2137–2148.

[17] K. Romer, Time synchronization in ad hoc networks, in: ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2001.

[18] J. Lee, S. Venkatesh, M. Kumar, Formation of a geometric pattern in mobile wireless sensor networks, Journal of Robotic Systems 21 (10) (2004) 517–530.

[19] M. Singh, V. K. Prasanna, Supporting topographic queries in a class of networked sensor systems, in: Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS), 2005.

[20] K. Nakano, S. Olariu, Energy-efficient randomized routing in radio networks, in: ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM), 2000, pp. 35–44.

[21] K. Nakano, S. Olariu, A. Zomaya, Energy-efficient permutation routing in radio networks, IEEE Trans. on Parallel and Distributed Systems 12 (6) (2001) 544–557.

[22] T. ElBatt, On the scalability of hierarchical cooperation for dense sensor networks, in: ACM/IEEE International Symposium on Information Processing in Sensor Networks, 2003.

[23] C. Florens, M. Franceschetti, R. J. McEliece, Lower bounds on data collection time in sensory networks, IEEE Journal on Selected Areas in Communication 1 (11) (2004) 1–10, special Issue on Fundamental Performance Limits in Sensor Networks.

[24] E. J. Duarte-Melo, M. Liu, Data-gathering wireless sensor networks: Organization and capacity, Computer Networks (COMNET) 43 (4) (2003) 519–537, special Issue on Wireless Sensor Networks.

[25] C.-F. Chiasserini, M. Garetto, Modeling the performance of wireless sensor networks, in: IEEE InfoCom, 2004.

[26] P. Leone, J. Rolim, Towards a dynamical model for wireless sensor networks, in: International Workshop on Algorithmic Aspects of Wireless Sensor Networks, 2004.

[27] T. J. Shepard, A channel access scheme for large dense packet radio networks, ACM SIGCOMM Computer Communication Review 26 (4) (1996) 219–230.

[28] D. Scherba, P. Bajcsy, Communication models for monitoring applications using wireless sensor networks, Tech. rep., the National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana-Champaign (2004).

[29] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, J.-P. Sheu, The broadcast storm problem in a mobile ad hoc network, in: ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 1999.

[30] B. Williams, T. Camp, Comparison of broadcasting techniques for mobile ad hoc networks, in: ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2002.

[31] K. Viswanath, K. Obraczka, Modeling the performance of flooding in multihop ad hoc networks, in: Symposium on Performance Evaluation of Computer and Telecommunication Systems, 2004.

[32] Y. Sasson, D. Cavin, A. Schipe, Probabilistic broadcast for flooding in wireless mobile ad hoc networks, in: IEEE Wireless Communications and Networking Conference (WCNC), 2003.

[33] A. Wang, S.-H. Cho, C. G. Sodini, A. P. Chandrakasan, Energy-efficient modulation and MAC for asymmetric microsensor systems, in: International Symposium on Low Power Electronics and Design (ISLPED), 2001.

[34] C. S. Raghavendra, S. Singh, PAMAS – power aware multi-access protocol with signaling for ad hoc networks, ACM SIGCOMM Computer Communication Review 28 (3) (1998) 5–26.

[35] J. Deng, B. Liang, P. K. Varshney, Tuning the carrier sensing range of ieee 802.11 mac, in: IEEE GlobeCom, 2004.

[36] GloMoSim Project.
URL `http://pcl.cs.ucla.edu/projects/glomosim`

A To Incorporate Carrier Sensing Range

When carrier sensing range is considered, it is assumed that a packet transmission to a given destination will be collided if another packet is transmitted at the same time from a sensor node within the carrier sensing range of the destination. Without loss of generality, we assume that the carrier sensing range is twice the transmission range.

Consider the case where K_1 nodes need to send a packet to a common destination in their transmission range in a time phase of s time slots while K_2 nodes in the carrier sensing range of the destination but not its transmission range also need to broadcast once within this s time slots. We calculate the probability for the destination to successfully receive at least one packet from any of the K_1 senders. Mathematically speaking, consider the problem of randomly dropping $K_1 > 0$ identical items of type A and $K_2 \geq 0$ identical items of type B into $s > 0$ identical buckets. We are interested in the probability of having at least one bucket to hold exactly one item of type A and none of type B. Based on a similar idea for deriving Equation (2a), such a probability can be obtained using the following recursive representation. Let $\mu'(K_1, K_2, s)$ denote the desired probability. We have:

$$\mu'(K_1,K_2,s) = \begin{cases} 1 & \text{if } K_1 = 1 \text{ and } K_2 = 0, \quad \text{(A.1a)} \\ K_1(\frac{(s-1)^{K_1+K_2-1}}{s^{K_1+K_2}}) + (\frac{s-1}{s})^{K_1+K_2}\mu'(K_1,K_2,s-1) + \\ \quad \sum_{i=2}^{K-1}\sum_{j=1}^{K_2}\binom{K_1}{i}\binom{K_2}{j}\frac{(s-1)^{K_1+K_2-i-j}}{s^{K_1+K_2}} \\ \quad \mu'(K_1-i,K_2-j,s-1) & \text{otherwise.} \quad \text{(A.1b)} \end{cases}$$

Now consider a node u in ring R_j with a distance of $x \in [0, r]$ from the inner boundary of R_j. Let $B(x, k)$ denote the area in ring R_k that is within distance r to $2r$ from u, i.e., nodes in $B(x, k)$ are within the carrier sensing range of u, but not the transmission range of u. For u in ring R_j, it is clear that $B(x, k)$ is not empty only for $k = j - 2, \ldots, j + 2$. Specifically, we have:

$$\begin{aligned} B(x,j-2) &= f((j-2)r, 2r, x+r) \\ B(x,j-1) &= f((j-1)r, 2r, x) - B(x,j-2) - A(x,j-1) \\ B(x,j) &= f(jr, 2r, x-r) - \sum_{l=j-2}^{j-1} B(x,l) - \sum_{l=j-1}^{j} A(x,l) \\ B(x,j+1) &= f((j+1)r, 2r, x-2r) - \sum_{l=j-2}^{j} B(x,l) - \sum_{l=j-1}^{j+1} A(x,l) \end{aligned}$$

$$B(x, j+2) = \pi(2r)^2 - \sum_{l=j-2}^{j+1} B(x, l) - \sum_{l=j-1}^{j+1} A(x, l)$$

Now suppose that u has not received the broadcasted information after T_{i-1}. The number of nodes in rings $R_{j-2}, \ldots, R_{j+2}$ that have received the information in T_{i-1} is given by $n_{j-2}^{i-1}, \ldots, n_{j+2}^{i-1}$. Assuming that these nodes are uniformly distributed in the three rings, we can calculate the expected number of such nodes in the carrier sensing range of u but not the transmission range of u that have received the broadcasted information in T_{i-1}, denoted as $h(x)$.

$$h(x) = \sum_{k=j-2}^{j+2} (n_k^{i-1} \frac{B(x, k)}{C_k}) \; . \tag{A.2}$$

Now, we can develop a similar recursive presentation of n_j^i as Equation (4):

$$n_j^i = \int_0^{2\pi} \int_0^{r} (r(j-1) + x) \mu'(g(x), h(x), s) \frac{\delta C_j - \sum_{k=1}^{i-1} n_j^k}{C_j} dx d\theta \; . \tag{A.3}$$

$\square$