Communication Models for Algorithm Design in Wireless Sensor Networks

Yang Yu, Bo Hong, Viktor K. Prasanna
Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089-2562
{yangyu, bohong, prasanna}@usc.edu

Abstract

With continuing advancements in sensor node design and increasingly complex applications for wireless sensor networks (WSNs), formal communication models are needed for either fair comparison between various algorithms or the development of design automation in WSNs. Toward such a goal, we formally define two link-wise communication models, namely, the Collision Free Model (CFM) and the Collision Aware Model (CAM). While programming under CFM resembles similarity with traditional parallel and distributed computation, performance analysis in CFM is often inaccurate or even misleading, due to its over-simplification on modeling packet collisions. On the other hand, by exposing low level details of packet collisions in CAM, we are able to model and analyze the behavior of the network in a more realistic, accurate, and precise fashion. To validate the above arguments, we study a basic operation in WSNs —- broadcasting, using the well-known simple flooding and probabilistic based broadcasting schemes. We present an analytical framework that facilitates an accurate and precise modeling and analysis for both schemes in CAM, in terms of two performance metrics including reachability and latency. Our analytical results indicate that the impact of node density on the two performance metrics can be minimized by carefully choosing the probability of broadcasting, implying a strong scalability of the probability based broadcast scheme. Moreover, our analysis is confirmed by extensive simulation results.

I. Introduction

Wireless sensor networks (WSNs) are ad-hoc self-organizing untethered networks of smart sensors characterized by severe energy resource constraints. State-of-the-art wireless communication in WSNs 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 WSN applications.

Contemporary approaches for design and optimization of sensor network applications are centered around manual customization of the network protocol stack. Given the energy constraints in the network and end users' desire for long lived systems, this network-centric philosophy is acceptable for simple data gathering and querying applications. However, this design methodology is simply not feasible for future large-scale sensor networks primarily because the large number of protocols executing concurrently in WSNs – positioning, topology maintenance, medium access control, time synchronization, calibration, error detection, routing, and the application-level functionality - make it very difficult to manually optimize the design, while simultaneously ensuring correct operation.

With continuing advancements in sensor node design and increasingly complex applications, interest in **design automation** of sensor network applications is inevitable.The objective is to eventually enable domain experts to design and analyze algorithms, and automatically synthesize programs for an abstract machine model of the underlying system, without requiring a knowledge of low level networking aspects of the deployment. In this paper, as a initiative step toward our grand vision of building a methodology for design automation, we examine two models for abstracting link-wise communication in WSNs. Such link-wise communication models lay the foundation above which system-wide communication and computation models can be developed.

In fact, most existing works for algorithm design in sensor networks either explicitly or implicitly assume their own link-wise communication model with different set of assumptions. However, the lack of formal definition and systematic composition of these assumptions and model specifications remains an obstacle for either fair

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

comparison between different works or the development of design automation methodologies based on unified and commonly accepted computation and communication models. The success of the PRAM family in parallel and distributed systems [1] as well as the model based design automation and application synthesis in reconfigurable computation [2]–[4] motivate us to develop communication models for WSNs in a similar line.

The main contributions of the paper are listed as follows:

1) We formally define two models that explore the tradeoffs between ease of programming, closeness to real-life implementation, and the power of providing accurate and precise performance analysis and prediction. These models are not new in themselves. The contribution of this paper is to formally define them with a comparison of their pros and cons for algorithm design in WSNs.

   Our first model, the Collision Free Model (CFM) is very powerful model in terms of programming by abstracting all details of low level channel contention and packet collision away from the algorithm designers. By abstracting reliable communication as an atomic operation, programming based on CFM bears resemblance to existing algorithm design in 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 not accurate or even misleading.

   Our second model, the Collision Aware Model (CAM) specifies that packet collisions occur when multiple senders try to communicate to their common neighbors at the same time. Hence, CAM closely models current technologies for wireless communication (such as the broadcast mechanism in 802.11). Since CAM shifts all the responsibility for detecting and handling packet collisions to the algorithm designer, reliable packet delivery in CAM requires extra programming efforts. However, the exposure of low level details also offers us better opportunities to (1) perform more accurate performance analysis, in terms of both functionality and energy-efficiency, and hence (2) design more efficient algorithms for WSNs. We illustrate this advantage using the example of broadcasting, as described in the next contribution.

2) 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 WSNs [5]–[7]. We consider two traditional performance metrics for evaluating various broadcast schemes [8]–[10]: (1) to maximize the reachability for a given a latency constraint; and (2) to minimize the latency for a given reachability constraint.

   We study two simple broadcasting schemes that are suitable for WSNs, simple flood and probability based broadcast. Though simple flooding fits into CFM very well, we argue that the information provided by CFM is not sufficient for a realistic and accurate performance analysis of simple flooding. However, by utilizing packet collision information in CAM, we present an analytical framework that is capable of precisely and accurately modeling the behavior of probability based broadcast, which includes simple flooding as a special case. Such analytical information helps the design of more efficient broadcasting, in terms of optimizing the above two performance metrics.

   Our analytical results indicate that for the two performance metrics, the optimal probability of broadcast decreases fast with node density. More importantly, the impact of node density on the two performance metrics can be minimized by carefully choosing a probability, implying a strong scalability of the probability based broadcast scheme. This observation is important to the design and implementation of time or energy efficient probability based broadcasting in practice. We also show that our analytical results match very well to our simulation results.

Our framework easily extends to model the energy cost of the studied broadcast schemes. Due to space limitation, we only present results for reachability and latency in this paper. Analytical and simulation results for modeling energy cost can be found in [11].

The rest of the paper is organized as follows. We briefly discuss related work in Section II. Our system assumptions and formal description of CFM and CAM are presented in Section III. The analysis of two simple broadcasting schemes under CFM and CAM is presented in Section IV. Finally, we give concluding remarks in Section VI.

II. Related work

Although a huge amount of literature exists about algorithms to perform various tasks on WSNs, little work has been dedicated to define clear models of the underlying communication or computation mechanisms. Nevertheless, the assumptions and models described in this paper have been widely used in various context.

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 [12], [13], data gathering [14],

[15], performance analysis [16], [17], and localization and time synchronization algorithms [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 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 is from a relatively high level perspective that facilitates efficient algorithm design in WSNs. Hence, low level details with respect to hardware devices, protocol implementation, and physical layer signal propagation are abstracted away. Many existing works have addressed such low level details [27], [28].

Since broadcasting is a basic operation in large scale networks, the behavior of broadcasting has been previously studied [8]–[10]. A nice categorization of broadcasting schemes were presented by Williams *et. al.* [9], which consists of simple flooding, probability based scheme, area based scheme and neighbor knowledge scheme. Since area based scheme requires localization knowledge for each node and neighbor knowledge scheme requires the knowledge about two-hop neighborhood, they are not suitable for WSNs where to obtain the required information is too expensive. Therefore, we focus on the simple flooding and probability based scheme in this study. While an analytical study of the reachability of simple flooding is given in [10], the performance of the probability based scheme has only been studied through simulation [8], [9]. In our study, we present an analytical framework to reveal the optimal probability of broadcasting for either maximizing reachability given a latency constraint or minimizing latency given a reachability constraint. Also, our framework is extendible for the area based scheme and neighbor knowledge scheme.

III. Our Models

A. System Assumptions

We make the following assumptions about the underlying network.

1) The system consists of a set of n homogeneous sensor nodes, V, with identical transmission radius r and the same time and energy costs for sending or receiving 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 is switched off when it does not need to participate in any communication. Hence, the energy cost of 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 hardware failure of nodes are not modeled.
6) While we do not assume time synchronization in the system, we assume that certain virtual time 0 is available for starting the execution of the application.
7) 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 symmetric communication assumed in Assumption 2 also serves the purpose of abstracting low level details for high level algorithm design. There two assumptions are widely used in papers listed in Section II.

Assumption 3 is widely used for distributed algorithm design and protocol implementation in WSNs [7]. 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 [29]. The time and energy cost of these mechanisms is assumed to be absorbed into the cost of normal communication.

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.

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. The virtual time 0 can be realized by broadcasting a strong beacon signal from the base station. 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.

Assumption 7 is also frequently used by many works that focus on algorithm design to model packet collision (as described in Section II). 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.

B. 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 our following descriptions.

1) Collision Free Model: 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 the assumption 1 in Section III-A, the same time and energy costs apply to both the sender and the receivers.

FCM is a very powerful model since it allows fully parallelized packet transmission, which significantly facilitates high level algorithm design. 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 in a given task. However, the cost of contention resolution shall be properly reflected by the above time and energy parameters, which is tightly related to the deployment of the network and the implementation of MAC layer.

FCM can be implemented in the following ways. A naive implementation based on 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. However, such techniques require additional hardware and more complicated coordination among sensor nodes, which is beyond the capability of contemporary sensor nodes. Hence, we consider the implementation by using acknowledgment only.

2) Collision Aware Model: 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.

IV. A Case Study: Broadcasting in Wireless Sensor Networks

The broadcasting problem considers the distribution of a piece of information from a source node to the whole network. For example, the source node can be the base station, where user queries are injected into the network. Since the broadcasting problem has been well studied in various context as described in Section II, we are not proposing any novel design schemes in this paper. Instead, we focus on two suitable schemes for WSNs.

The first scheme is called *simple flooding*, where each node broadcasts exactly once after it receives the information from any neighbor for the first time. While simple flooding fits perfectly into CFM where packet collision is abstracted away, various studies have revealed that the performance of this scheme in real-life systems where packet collisions cannot be ignored degrades severely from its analytical prediction in CFM [8]. In other words, the performance prediction of simple flooding in CFM is often inaccurate and even misleading. Hence, we also study another scheme modified from simple flooding by forcing each node to broadcast with a pre-specified probability, referred to as the *probability based broadcasting*. It is clear that simple flooding is a special case of probability based broadcasting by setting the probability of broadcasting to exactly 1. While CFM does not provide sufficient information for a meaningful performance modeling of the probability based broadcasting, the exposure of low level packet collisions in CAM enables a realistic and accurate performance analysis for this scheme.

To focus on the main issue, we consider the time and energy costs of communication only for the studied algorithms in our analysis and omit the cost of computation.

A. Performance Metrics

Two performance metrics have been previously studied, namely *reachability* and *latency* [8]–[10]. In their traditional definition, 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 WSNs, 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 six performance metrics:

1) Given a latency constraint, maximize the reachability.
2) Given a latency constraint, minimize the energy cost.
3) Given a reachability constraint, minimize the latency.
4) Given a reachability constraint, minimize the energy cost.
5) Given an energy constraint, maximize the reachability.
6) Given an energy constraint, minimize the latency.

The second and the last metrics are not meaningful, since both of them leads to the trivial solution that no broadcast should be performed. The rest 4 metrics are used to evaluate a broadcasting scheme. Intuitively, to maximize reachability given a latency constraint (the first metric) is a dual problem of minimizing latency given a reachability constraint (the third metric); to minimize energy cost given a reachability constraint (the forth metric) is a dual problem of maximizing reachability given an energy constraint (the fifth metric). While these 4 performance metrics are trivial for simple flooding in CFM, no formal analysis has been performed for the probability based broadcasting in CAM. Due to space limitation, we show in this paper that our analytical framework can be used to determine the best probability to optimize the first and third metrics. Results for optimizing the forth and fifth metrics can be found in [11].

To focus on a general and yet realistic network scenario as well as make the analysis tractable, we consider a network with n nodes uniformly distributed in a circle of radius kr, where k is a pre-specified parameter. We also assume that the source node is placed at the center of the circle. Since the scalability of various broadcasting scheme is of particular importance from algorithm design perspective, we use node density as the only system configuration parameter.

B. Simple Flooding in CFM

Since each broadcast is guaranteed to succeed in CFM, the optimization of the 4 performance metrics in Section IV-A becomes trivial. For the considered network with n sensor nodes within a circle of radius kr, it is easy to see that simple flooding achieves a reachability of 1 with time cost $O(kt_f)$ and energy cost $O(ne_f)$ (since each node needs to receive at least one broadcast and then broadcast once).

The above performance analysis essentially abstracts all information about the underlying network deployment, including node density, into the two quantities, t_f and e_f. Although the value of t_f and e_f relies heavily on node density, unfortunately, their inter-relationship is not reflected in CFM. Hence, it is not concrete and accurate enough to predict the performance of flooding purely based on t_f and e_f. For example, various studies have shown that it is very difficult for simple flooding to achieve a reachability of 1 in real networks [8], [10]. In other words, the impact of node density on packet collision has to be explicitly modeled for a realistic performance analysis, which is simply not doable in the over-simplified CFM.

C. Probability Based Broadcasting in CAM

Let p denote the probability of broadcasting at any nodes after it receives the broadcasted information from a neighbor node for the first time. As previously described, by setting $p \in (0,1]$, we can study the performance of both general probability based broadcasting and simple flooding. In the following, we present our analytical framework for modeling the performance of 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 consists 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 [9] 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. 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 k 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.

Before proceeding with the analysis, we define two notations. Consider two intersecting circles L_1 with radius R_1 and L_2 with radius R_2 (Figure 6). Let x denote the distance between the center of L_2, u and the border of L_1 (x is negative if u is inside L_1). Let $f(R_1, R_2, x)$ denote the area of intersection. Also, we consider the case where K nodes need to send a packet to a common destination in a time phase of s time slots. Let $\mu(K, s)$ denote the probability for the destination to successfully receive at least one packet from any of the K senders. The calculation of $f(R_1, R_2, x)$ and $\mu(K, s)$ are presented in Appendix I.

D. Modeling the Performance of PB_CAM

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

Let δ denote the density of the network, i.e., the average number of nodes in unit area. We assume strictly aligned time phases, with $T_1, T_2, \ldots$ denote the time phases from virtual time 0. 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 $n_j^i = 0$ for any other combination of $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots$.

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 are 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:

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

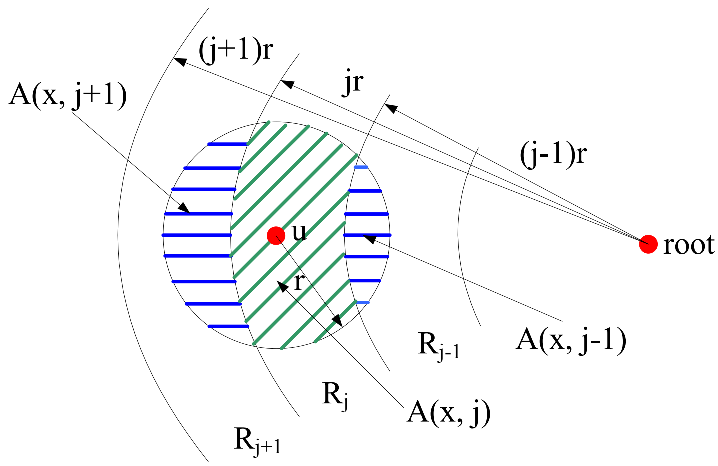


Fig. 1. The partition of the communication range of u

These three portions of area actually form a partition of the area covered by u's communication radius (Figure 1). 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^i_{j-1}, n^i_j, and n^i_{j+1}. Assuming 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, denoted as $g(x)$. We have

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

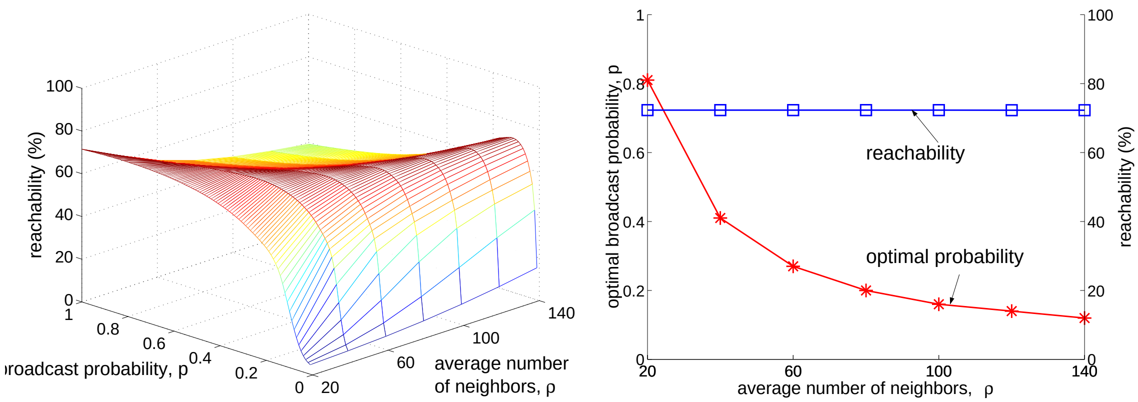
Recall that p is the probability for each node to broadcast. The probability that u will successfully receive at least one packet from the $g(x)$ number of nodes is $\mu(g(x)p, s)$. Since we have assumed that the nodes in ring R_j that have received the information after T_{i-1} are uniformly distributed in R_j, the nodes in R_j that have not received the information also follow a uniform distribution. Hence, by integrating $\mu(g(x)p, s)$ over the area of R_j, we can derive the expected number of nodes in R_j that will receive the information during T_i. Hence, we have

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

For a given i, the above representation of n^i_j can be recursive applied over $j = 1, 2, \ldots, k$ to calculate the number of nodes in each ring that receive the information after the i-th time phase. Therefore, given a network confi guration in terms of k, s, and node density δ, we are now able to model the behavior of PB_CAM for the optimization of the 4 performance metrics described in Section IV-A.

1) Maximize Reachability for A Given Latency Constraint: To incorporate the effect of the communication radius r, the node density is captured by the average number of neighbors within the communication range of a node, denoted as ρ. That is, $\rho = \pi r^2 \delta$ (ignoring the boundary effect). We consider a network with $k = 5$ 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 set $s = 3$ and p from 0.01 to 1 in increments of 0.01. In Figure 2(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.

In Figure 2(b), we plot the optimal probability 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 broadcasting probability, implying a very good scalability of



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

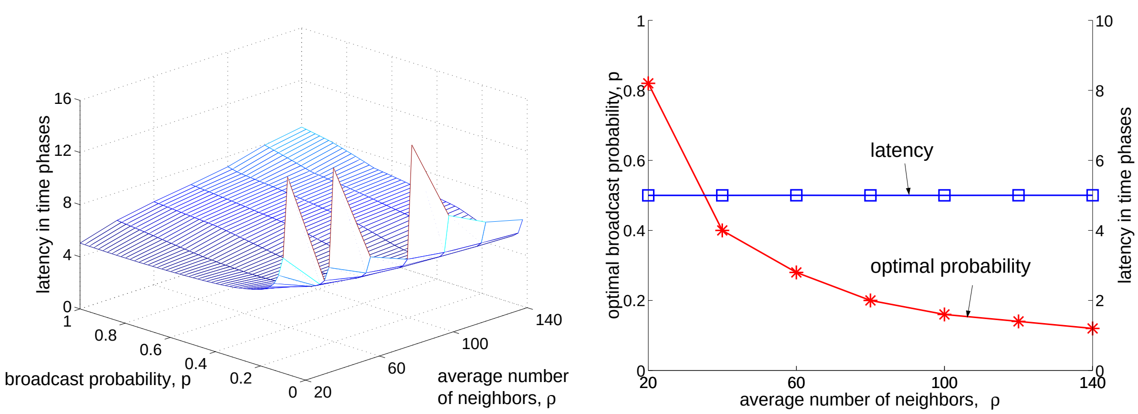
(b) Optimal probability with the corresponding reachability

Fig. 2. Reachability of PB_CAM in 5 time phases

PB_CAM with respect to node density. As a comparison, the curve with probability 1 in Figure 2(a) is actually the reachability of simple flooding in CAM, which is around 0.55 of the optimal when $\rho = 140$.

2) Minimize 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 instead of discrete values of time phases.

In Figure 3, we show the latency in terms of number of 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.



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

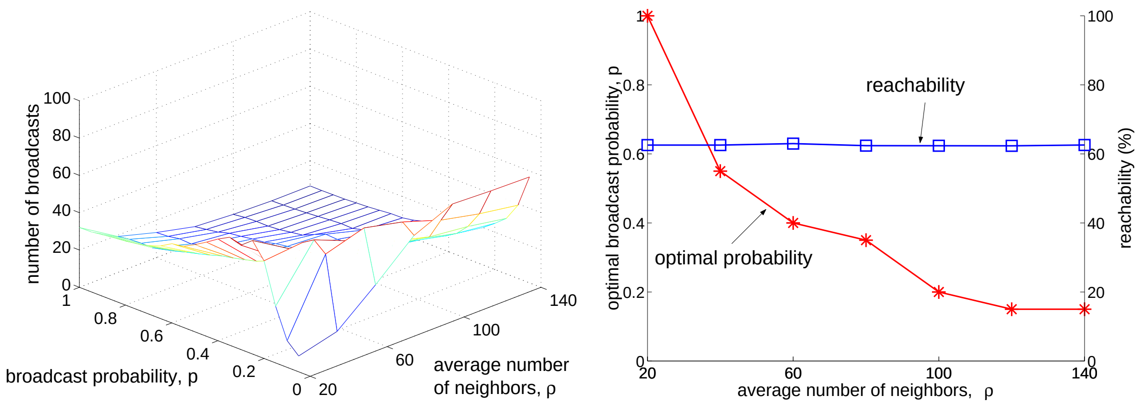
(b) Optimal probability with corresponding latency

Fig. 3. Latency of PB_CAM for achieving 72% reachability

From Figure 3(b), we observe that regardless of the variations of ρ, the 72% reachability can be achieved in 5 time phases by carefully adjusting the broadcasting probability. However, from Figure 3(a), simple flooding (the curve with $p = 1$) uses more than 8 time phases when $\rho = 140$. Also, the optimal probability curve in Figure 3(b) is the same as the one in Figure 2(b). This is understandable, since to minimize latency for a given reachability constraint is actually a dual problem of maximizing reachability given a latency constraint.

V. Simulations and Verifications

To validate our results in Section IV-B, we performed extensive simulations using the framework provided by GlomoSim. The parameter settings for our simulation were kept the same as those used for the analysis in Section IV-D, except that for a reasonable running time, we varied broadcasting probability from 0.05 to 1 in increments of 0.05. The data shown in this section is averaged over 30 random runs.

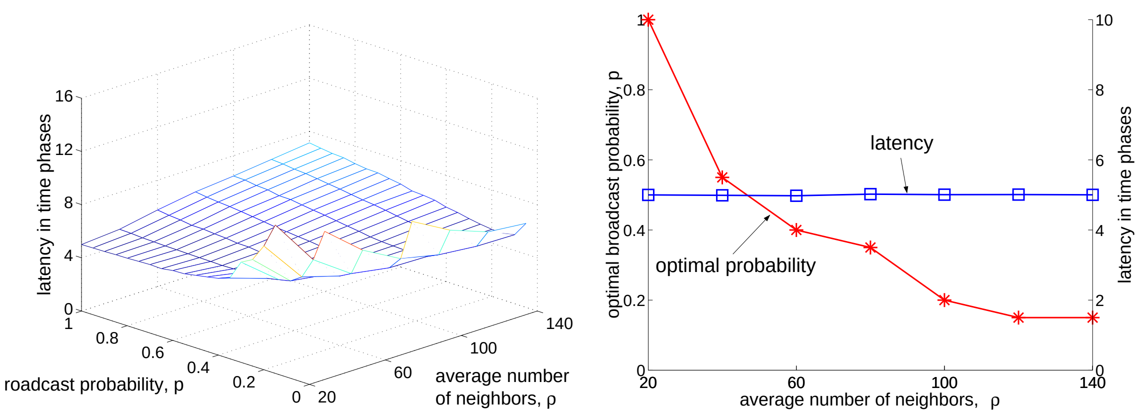


(a) Reachability

(b) Optimal probability and corresponding reachability

Fig. 4. Simulation results of the reachability of PB_CAM in 5 time phases

We first illustrate the reachability for 5 time phases in Figure 4. It can be observed that our analytical results match the simulation results in Figure 2 quite well. The optimal probability decreases with a similar trend as the curve in Figure 2(b). Also, the achievable reachability is consistently around 63% with respect to variations in ρ.



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

(b) Optimal probability and corresponding latency

Fig. 5. Simulation results of the latency of PB_CAM for 63% reachability

In Figure 5, we show the latency to achieve a reachability of 63%. It can be observed that the optimal probability is the same as the optimal probability in Figure 4(b) and the corresponding latency is 5 time phases. This also confirms our analytical results in Figure 3.

VI. Concluding Remarks

We have illustrated the pros and cons of two link-level communication models for algorithm design in wireless sensor networks, namely, the Collision Free Model (CFM) and the Collision Aware Model (CAM). While CFM facilitates high-level algorithm design and performance modeling, its over-simplification on packet collision does not provide much meaningful interpretation of its performance prediction in real-life systems. However, by exposing details of packet collision, CAM can be used to accurately and precisely model the performance of a probability based broadcasting scheme.

Specifically, we have used our analytical framework to model the behavior of probability based broadcasting in CAM, PB_CAM. Both our analytical and simulation results indicate two important factors: (1) the optimal probability to either maximize reachability within a latency constraint or minimize latency to satisfy a reachability constraint strongly depends on node density; and (2) by carefully choosing the probability of broadcasting, PB_CAM scales very well with respect to node density.

REFERENCES

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

[2] J. C. Cogolludo and S. Rajasekaran, "Permutation routing on reconfigurable meshes," *Algorithmica*, vol. 31, no. 1, pp. 44–57, 2001.

[3] R. Lin, "Reconfigurable parallel inner product processor architectures," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 9, no. 2, pp. 261–272, Apr. 2001.

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

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

[6] C. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," in *IEEE WMCSA*, Feb. 1999, pp. 90–100.

[7] C. Intanagonwiwat, R. Govindan, and 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.

[8] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and 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.

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

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

[11] Y. Yu, B. Hong, and V. K. Prasanna, "Communication models for algorithm design automation in wireless sensor networks," University of Southern California, Tech. Rep., 2004. [Online]. Available: http://halcyon.usc.edu/~yangyu/data/TR_CENG_200420.pdf

[12] B. Hong and 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)*, Oct. 2004.

[13] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Annual Symposium on Foundations of Computer Science (FOCS)*, 2003.

[14] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On network correlated data gathering," in *IEEE InfoCom*, 2004.

[15] M. Enachescu, A. Goel, R. Govindan, and R. Motwani, "Scale free aggregation in sensor networks," in *1st International Workshop on Algorithmic Aspects of Wireless Sensor Networks (Algosensors)*, 2004.

[16] Z. Cheng and 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.

[17] S. Pattem, B. Krishnamachari, and 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.

[18] N. Patwari, A. O. H. III, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Transactions on Signal Processing, Special Issue on Signal Processing in Networks*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.

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

[20] K. Nakano and S. Olariu, "Energy-efficient randomized routing in radio networks," in *ACM DIALM*, 2000, pp. 35–44.

[21] K. Nakano, S. Olariu, and A. Zomaya, "Energy-efficient permutation routing in radio networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 12, no. 6, pp. 544–557, June 2001.

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

[23] C. Florens, R. J. M. Eliece, and M. Franceschetti, "Data collection strategies for sensory networks," *IEEE Journal on Selected Areas in Communication. Special issue on fundamental performance limits in sensor networks*, 2004.

[24] E. J. Duarte-Melo and M. Liu, "Data-gathering wireless sensor networks: Organization and capacity," *Computer Networks (COMNET) Special Issue on Wireless Sensor Networks*, vol. 43, no. 4, pp. 519–537, Nov. 2003.

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

[26] P. Leone and 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 ComputerCommunication Review*, 1996.

[28] D. Scherba and P. Bajcsy, "Communication models for monitoring applications using wireless sensor networks," NCSA, UIUC, Tech. Rep., 2004.

[29] C. S. Raghavendra and S. Singh, "PAMAS – power aware multi-access protocol with signaling for ad hoc networks," *Computer Communication Review*, July 1998.

APPENDIX I
THE CALCULATION OF $f(R_1, R_2, x)$ AND $\mu(K, s)$

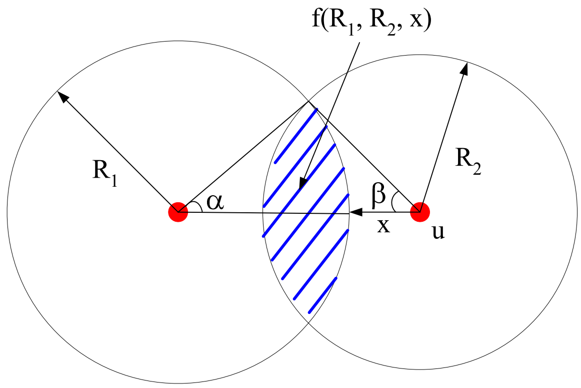


Fig. 6. The intersection of two circles

We first consider the scenario where two circles L_1 and L_2 intersect with each other (Figures 6) and calculate the area of intersection. Let R_1 denote the radius of L_1, and R_2 denote the radius of L_2. Let x denote the distance between the center u of L_2 and the border of L_1. We set x to be positive when u is outside L_1, and negative when u is inside L_1. We have

$$\alpha = \arccos(\frac{R_1^2 + (R_1 + x)^2 - R_2^2}{2R_1(R_1 + x)}), \text{ and } \beta = \arccos(\frac{R_2^2 + (R_1 + x)^2 - R_1^2}{2R_2(R_1 + x)}) \ .$$

Hence, the area of intersection, denoted as $f(R_1, R_2, x)$ can be calculated as:

$$f(R_1, R_2, x) = \alpha R_1^2 - R_1^2 \sin\alpha \cos\alpha + \beta R_2^2 - R_2^2 \sin\beta \cos\beta \ . \tag{3}$$

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. Such a probability does can be solved using the following recursive representation. Let $\mu(K, s)$ denote the desired probability. We have:

$$\mu(K, s) = \begin{cases} 1 \text{ if } K = 1 & \text{(4a)} \\ K(\frac{(s-1)^{K-1}}{s^K}) + (\frac{s-1}{s})^K \mu(K, s-1) + \sum_{m=2}^{K-1} \binom{K}{m} \frac{(s-1)^{K-m}}{s^K} \mu(K-m, s-1) & \text{(4b)} \end{cases}$$

The rationale behind the above equation is that if we have exactly one item, 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;
- No item is dropped in the first bucket;
- m items are dropped in the first bucket, where $m = 2, \ldots, K-1$.

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.