

# Belief Propagation in Wireless Sensor Networks - A Practical Approach

Tal Anker<sup>1,2</sup>, Danny Dolev<sup>1</sup>, and Bracha Hod<sup>1</sup>

<sup>1</sup> The Hebrew University of Jerusalem, Israel

{anker,dolev,hodb}@cs.huji.ac.il

<sup>2</sup> Marvell Semiconductor, CA, USA

tala@marvell.com

**Abstract.** Distributed inference schemes for detection, estimation and learning comprise an attractive approach to Wireless Sensor Networks (WSNs), because of properties such as asynchronous operation and robustness in the face of failures.

Belief Propagation (BP) is a method for distributed inference which provides accurate results with rapid convergence properties. However, applying a BP algorithm to WSN is challenging. Many papers that proposed using BP for WSNs do not consider all of the constraints which these networks impose.

This paper presents a framework that implements both localized and data-centric approaches to improve the effectiveness and the robustness of this algorithm in the WSN environment. The proposed solution is empirically evaluated, as applied to the clustering problem, and it can be easily extended to suit many other applications that use BP as an underlying algorithm.

**Keywords:** Belief Propagation, Wireless Sensor Networks.

## 1 Introduction

It is generally believed that Wireless Sensor Networks (WSNs) will be ubiquitously accepted as an infrastructure for applications in areas as diverse as environmental monitoring, health-care applications, and home automation. Data fusion and processing will be the core information gathering activities performed by the sensor nodes. Consequently, inference methods, which are important means of performing data fusion, have become an increasing research interest in the field of WSNs.

The goal of distributed inference in WSNs can be achieved using several methods that were originally developed for graphical models [1], such as Belief Propagation (BP) [2]. BP is an iterative algorithm for computing maximal or marginal posterior probability, by means of local message-passing. BP is presented in the literature as an effective and useful inference method for a wide range of communication applications and network topologies, including WSNs [3], [4].

The adoption of an inference algorithm such as BP for WSNs presents a great opportunity, because learning techniques which fully utilize the available

information can achieve nearly optimal results. However, it is also a formidable challenge, due to the distinctive characteristics of these networks: Energy efficiency is a major design goal in WSNs because the nodes have limited power sources and restricted computational capacities. The wireless medium imposes many other constraints, such as collisions and errors. Other properties of this medium, such as interferences and poor link quality, result in changes to the topology of the network, which, together with the fact that the network is self-organizing, create a unique and challenging network dynamic. WSNs are also likely to have a large number of nodes which may result in potentially drastic scaling problems.

The significance of this paper is twofold:

Firstly, our research provides important insights regarding real-world challenges in WSNs, which may significantly affect the inference quality. These insights are relevant to many WSN inference schemes, and are addressed in this paper in a broader manner than has been previously presented (such as in [3], [4], [5] and [6], which either posit impractical assumptions about network topology, or otherwise neglect issues such as the overhead of communication or scalability of the solution).

Secondly, driven by the need for a practical solution, we propose a general BP framework that takes both localized [7] and data-centric [8] approaches. Using simulations, we demonstrate and analyze the properties of the scheme within the context of a solution to the clustering problem. The entire framework is fully distributed and localized, and presents an excellent approximation to the optimal inference solution. Moreover, it is shown to be asynchronous and robust, and to introduce only a minor and consistent cost in communication and overhead, regardless of the size of the network.

In contrast with previous work in this area [9], this paper focuses on the general construction and properties of the BP framework in the WSN, rather than on the application itself. The goal of this paper is to understand the general characteristics of the BP framework, which has not been presented to date. Introduction of new concepts about the clustering task is beyond the scope of the paper.

The rest of the paper is organized as follows. Section 2 briefly describes the related work in this area. A short background is provided in Section 3. The practical issues involved in WSNs are described in Section 4. An efficient BP scheme for distributed inference is presented in Section 5. Section 6 includes an analysis of this method using simulations. Section 7 concludes the paper with a summary.

## 2 Related Work

A graphical-model-oriented perspective of distributed data fusion in WSNs is presented in [10]. The paper provides a bridge between the field of graphical models to the data fusion in WSNs, discusses the tradeoffs between approximation and energy conservation and presents message censoring as an approach for solving the problem. The paper concludes that the results are far from complete and that the mapping between the two domains is still an area for research.

A general and robust architecture for distributed inference in sensor networks is proposed in [5]. The architecture presented in that paper considers practical issues and provides an analysis using a real deployment of WSNs. The method is based on a junction tree for message-passing, and as such has two main drawbacks: First, construction and maintenance of the tree require a large amount of communication and processing overhead, as well as usage of reliable mechanisms. Moreover, exact inference in large networks becomes unrealistic since the method scales exponentially with the number of nodes, because of the complexity of the junction tree.

Loopy belief propagation (LBP) [11] is presented in [3] as an attractive method for use with WSNs, due to its distributed nature and its robustness in environments with asynchronous communication, noise and failures. However, the paper in question does not deal with practical issues, such as energy consumption or topology changes. Furthermore, the LBP was proven to be effective mainly in decoding applications, when the graph has long cycles. This is not the situation in WSNs, which contain many short cycles.

Nonparametric belief propagation is proposed in [4] for solving the localization problem. That paper is the first to present the broadcast variation of BP and it refers to many of the communication constraints that might appear in sensor networks. While that paper uses the LBP, our alternative approach of operating on trees can suit other applications. Additionally, some realistic issues, such as the effect of topology changes, are beyond the scope of that paper.

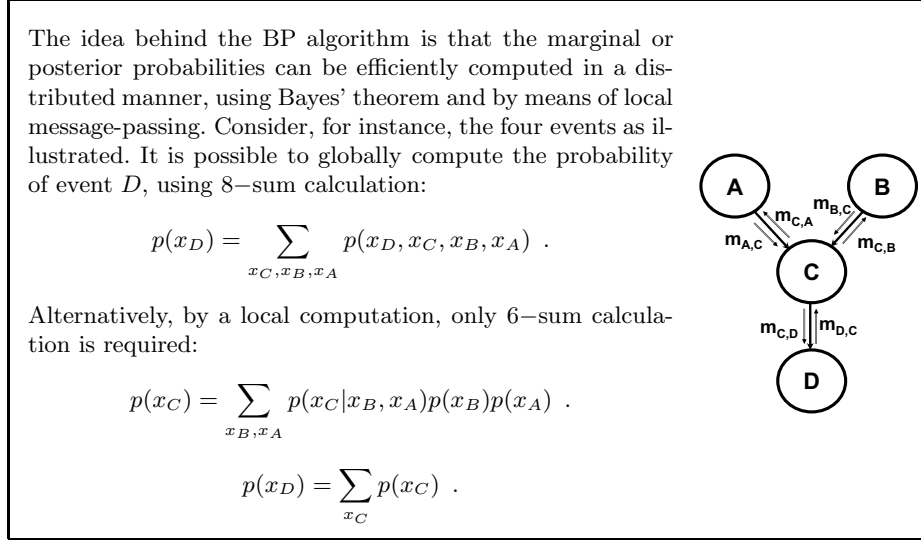
Reweighted belief propagation, implemented by [6], simulates a running of the basic algorithm of BP multiple times on different spanning trees, using different weights each time, to overcome the convergence problem of LBP and to find a fixed point. As presented in [10], the amount of communication required for the Tree Reweighted Max Product method is significantly larger than the basic max product algorithm.

### 3 Background

Graphical models [1] play an important role in machine learning algorithms that deal with uncertainty and complexity. They involve a mixture of probability theory and graph theory and are based on the basic idea of modularity, thus allowing a complex system to be viewed as a combination of many simpler pieces connected by probability theory. The graph theoretical aspects of the models provide a methodology to understand and formulate the system.

In a probabilistic graphical model, an undirected graph  $G = (V, E)$  is a set of nodes  $V$  and arcs  $E$ , which represent dependencies among random variables. We denote by  $x_i$  the variable representing the set of possible states of a node  $i$ .  $\psi_i(x_i)$  represents a local (previously known) distribution function of node  $i$  and  $\psi_{ij}(x_i, x_j)$  refers to a joint function of two connected nodes  $i$  and  $j$ . These functions are also called potential functions.

In the BP method [12], [13], the inference is carried out in a local and distributed manner by each node, using a message-passing technique.  $m_{ij}(x_j)$  is a



**Fig. 1.** Intuition to the BP method

message from node  $i$  to node  $j$  regarding the next state of node  $j$ . Node  $i$  calculates the message using previous messages it receives from its adjacent neighbors  $N(i)$ . The message update rule performed by a node  $i$  in round  $t$  is:

$$m_{ij}(x_j)^t = \sum_{x_i} \psi_i(x_i) \psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{ki}(x_i)^{t-1} .$$

The update rule being calculated by node  $i$  to determine the preferred state  $x_j$  of node  $j$  is a sum of all the possible states  $x_i$  of node  $i$ , assuming that  $j$  is in state  $x_j$ . Three elements are incorporated to each state: previously known information about the local node  $\psi_i(x_i)$ , the joint function  $\psi_{ij}(x_i, x_j)$  and the information about the immediately adjacent neighbors  $m_{ki}(x_i)^{t-1}$ .

Upon termination, after round  $\bar{t}$ , the belief at a node  $i$  (the marginal of the variable) is the product of the local evidence together with all the incoming messages and a normalization constant  $\alpha$ :

$$b_i(x_i) = \alpha \psi_i(x_i) \prod_{k \in N(i)} m_{ki}(x_i)^{\bar{t}} .$$

The BP algorithm for trees is an exact inference algorithm, which means that the belief converges to the correct marginal values in a finite number of iterations equal to the diameter of the tree.

A traditional BP algorithm massively uses floating point computations, which are expensive. We propose to use the Min-Sum algorithm [14], which is a variant of the BP algorithm, applied in the log-probability domain. The Min-Sum algorithm requires only addition and subtraction operations and works well with

integer values. The goal of the algorithm is to minimize the overall cost over all the nodes in the network, based on the local cost functions and the constraints among the nodes. The reader is referred to [9] for further details about the Min-Sum algorithm and its broadcast implementation.

## 4 Practical Issues

WSNs operate under a set of unique constraints and requirements that demand significant improvements and modifications to traditional algorithms. In particular, the BP algorithm cannot be embedded into WSNs in its original form. Several issues should be addressed to enable its efficiency in WSNs.

### 4.1 Mapping WSN to Graphical Model

Mapping of a WSN into the graphical model appears to be the most challenging task in the realization of this goal. The mapping of the network can be either to a tree or to a cyclic network.

Several papers, such as [3], have shown good results of the LBP in practical applications, mainly decoding. [15] discusses the convergence of the BP algorithm in general networks with single or multiple loops. The LBP worked well in these cases mostly because the cycles in the graph were large, so the effects of the cycles faded after only a few iterations. WSNs are associated with short cycles made by the broadcast range, so LBP is not an appropriate method for many applications in such networks. The use of the Min-Sum algorithm for energy efficiency also implies some limitations upon operating in a cyclic network. The convergence problem of the Min-Sum algorithm is similar to the convergence of the distance-vector routing protocol [16] and it is guaranteed only in acyclic networks. The split horizon rule cannot be applied in a BP algorithm, because of the algorithm structure.

Alternative methods remove the loops by replacing the cyclic network with trees. The junction tree is the most common such method and it is based on two properties: (1) every clique of the original graph is contained in some clique of the junction tree; and (2) for each node of the original graph, the cliques and all the edges containing it form a connected subtree of the junction tree.

Paskin and Guestrin in [5], argue about the need for a special architecture for distributed inference. Their major claim is that an optimized junction tree may reduce the overall communication cost. Additionally, the junction tree is more stable, flexible and does not depend on the network layer. The key disadvantage of the junction tree method is the large communication overhead it requires. Construction and maintenance of such a tree, when the network is continuously changing, incurs considerable overhead by the nodes.

### 4.2 Robustness against Failures

BP has a rapid convergence property, but when too many errors are involved, it is likely that the convergence will be slower and the nodes will converge to an

incorrect value. WSNs are exposed to a fairly large amount of communication and node failures. Apart from the ordinary failures in WSNs, such as packet errors and loss due to interferences and poor link quality, BP is especially vulnerable to broadcast message-passing, synchronization problems and topology changes during the message transmission.

**Broadcast Communication.** The wireless medium allows transmission of a single one-to-many message instead of multiple one-to-one messages. However at the same time, it imposes larger constraints on the shared medium, such as collisions and contention.

Message transmission in a broadcast manner reduces the communication volume, but at the same time it is much more sensitive to synchronization problems and is more error-prone than the original algorithm. Instead of receiving a unicast message  $m_{ji}(x_i)^t$  from node  $j$ , node  $i$  is required to extract the relevant information from the broadcast message, by a subtraction of its own information from the previous round's, and by calculating it on its own [9]:

$$m_{ji}(x_i)^t = \min_{x_j} \{ \psi_{ij}(x_i, x_j) + m_{j*}(x_j)^t - m_{ij}(x_j)^{t-1} \} .$$

The separation of a single update rule into two rules performed by different nodes, increases the potential errors and the synchronization issues that are involved with this method. Consider a situation where node  $j$  sends a message  $m_{j*}(x_j)^t$ . Upon reception of this message by node  $i$ , it subtracts its last message  $m_{ij}(x_j)^{t-1}$ , assuming that it was included in node  $j$ 's message. In the event of message loss, when  $m_{i*}(x_i)^{t-1}$  was not received by node  $j$  and was not included in its broadcast message, node  $i$ 's belief will be wrong and this error may be propagated through the network to other nodes. The nodes may ultimately converge to a common belief, but there is no guarantee that they will converge to the correct value. In the original protocol such a scenario will not occur, since the entire calculation is an atomic operation by a single node. In case of message loss, the nodes may synchronize in a subsequent iteration.

**Synchronization.** Perfect synchronization among the nodes in WSNs is difficult to achieve in practice, because of clock drifts. Therefore, message-passing would be better off if performed asynchronously, upon message reception from other nodes or upon external events.

The nodes' duty cycle is another factor to consider in the context of asynchronous operation; the message-passing algorithm should take into account cases where nodes wake up only in the middle of a process.

The general BP algorithm enforces some message ordering in each of the message-passing iterations. In the asynchronous method, there are no sequencing constraints and the messages may be transmitted arbitrarily during an iteration. Every node stores the received messages and computes them at the end of the iteration. The lack of synchronization thus introduces the additional cost of storage, and adds even further cost because some messages may be recomputed and retransmitted several times.

**Topology Changes.** WSNs are usually defined as semi-static networks because the nodes are not mobile in the sense of mobile networks. However, the nodes may nevertheless be repositioned by external factors, such as wind. Most commonly, the topology might change because the wireless links are not stable and sensor nodes are prone to failure. Therefore, the message-passing algorithm must not assume static topology during its invocation, and scenarios such as link break must be taken into consideration during the message-passing. A link break between some key node and its descendant may harm the convergence of the entire network, as the connectivity may be broken into separate components. Even when the message-passing tree is re-constructed, the synchronization between the nodes may not be restored. Therefore, it is very important to build a stable tree to minimize the effect of topology changes on the message-passing process, while managing such common scenarios.

### 4.3 Scalability

Since scalability is a main concern in WSNs, localized algorithms [7] are used as the building blocks in these networks. These localized algorithms are distributed and only a subset of the nodes participate. The nodes interact with each other only in a restricted vicinity, thus using only a limited amount of communication, computation, and storage resources - all crucial for energy efficiency in WSNs. While this approach seems to promise scalability, the design of such algorithms under the constraints of WSNs is not a trivial undertaking. Following this paradigm and the self-organization property of WSNs [17], the key challenge is to find localized behavior rules that may lead to the desired global property or at least approximate it, when applied by all the nodes.

Although BP is based on local message-passing, it is not inherently limited to a small region, and most of the proposed inference approaches based on BP are not localized. Localizing BP means that the algorithm is required to involve only part of the network and have a constant number of iterations, independent of the network diameter. Consequently, this decreases the number of transmitted messages and the time to deliver them, as well as resulting in low latency, regardless of the size of the network.

## 5 Efficient BP Framework for WSN

In light of the challenges presented in the previous section, this section describes the BP scheme for distributed inference in WSN.

### 5.1 Mapping WSN to Graphical Model

Our scheme maps a WSN into a graphical model by constructing multiple trees, where each tree combines the properties of the routing tree (such as hop count and link quality) together with clique properties. Thus, nodes that exist in the same clique in the graph are likely to be in the same message-passing tree. The

cliques in the graph may be generated according to some metric, depending on the application, and/or according to the physical layout. The spatial locality property of WSNs means that nodes which are physically close are likely to maintain relevant information, so it is common to have trees which were developed as a function both of their physical properties (e.g. a routing tree) and of the information which they contain.

The requirement for associating a clique to the tree, in addition to the routing requirements, can be understood from two different points of view. From the perspective of the graphical model, the ideal mapping of the network to a tree is to apply methods (such as the junction tree) which preserve the clique structure of the original graph. Construction of a tree, based on partial knowledge of the cliques in the graph results in a closer approximation of the actual junction tree, implying an improved result.

The second viewpoint is based on the data-centric approach to WSNs. Data centrality [7], [8] is a basic term in WSNs, which refers to the greater reliance upon the information content than on the geography or the identity of the nodes in the network. Concentration of the data content enables design of a more robust application, and outperforms idealized traditional schemes.

We improve our message-passing tree by considering the information that the nodes hold, similarly to the concept presented in the Directed Diffusion method [18]. Every tree is created on-the-fly using a single message that contains routing information, including parent and hop count, in addition to application-specific information. The fact that the tree is dynamically and locally created without any maintenance requirements, means that it scales and is efficient. The node that starts the inference process (i.e. with no prior information from its neighbors) operates as a root, by setting its hop count to zero. The nodes that receive the message can either select the sender as a parent, or wait a random short period (limited by a timer) in search of a better candidate. To be selected, a parent must fulfill the routing requirements and reside in the same clique in the graph. If a node does not find any parent after a given period, it operates as a root.

Each time the node selects a parent, it increments its hop count. This mechanism is used to detect and break cycles in the graph. Once every node is either designated as a root or has a parent, the trees are defined and it is now possible to perform the entire Min-Sum algorithm.

## 5.2 Robustness

The overall robustness of the algorithm has been presented. It should be noted that it is not possible to totally overcome the algorithm's sensitivity to failures, such as malformed messages and message loss. However, it is possible to reduce the occurrence of failures by using several heuristics:

1. The asynchronous nature of the sensors can be overcome by means of a "round" field in each message. This field designates the time interval in which messages are grouped together. Messages that arrive too early can be stored in a buffer and messages that arrive too late can be ignored.



**Tree Construction:**

- 
- 
- (1) Upon a triggering event or a timer:
    - (1.1) If no BP messages with positive *propagationLimit* have been received, start the process as a root by setting *hop* to zero and the localized predefined value of *propagationLimit*;
    - (1.2) Otherwise, select the best possible parent and start the process with the parent's *propagationLimit* decreased by one; The parent is defined as "final" if it meets all the requirements;
  - (2) Upon reception of a first-round BP message from other nodes:
    - (2.1) If already in the message-passing process:
      - (2.1.1) If the current parent meets both the routing tree requirement and the clique requirement  $\rightarrow$  process the message if it originates from this node's parent or descendant;
      - (2.1.2) If the current parent is not final: if the message's sender meets the requirements and also has a positive *propagationLimit*, then replace the parent with the message sender and process its message;
    - (2.2) If not in the message-passing process:
      - (2.2.1) If the message's sender meets all the requirements, and also has a positive *propagationLimit*, select that node as a parent and start the process with the given *propagationLimit* decreased by one;
      - (2.2.2) Otherwise, set a timer to start the process in a later time.
- 
- 

**Fig. 2.** Sketch of the Tree Formation

2. The "round" field in each message can also be used for detection and reproduction of message loss. Reproduction of the last message is performed by processing the last message that was received by this node, as if it had been received in the current round. Reproduction of the message keeps the nodes synchronized and enables convergence in later iterations.
3. The Min-Sum algorithm computes cost information by subtracting previous messages, under the assumption that the cost cannot decrease from one round to another. We use a broadcast version of this algorithm, which can cause errors in the subtraction operation, in that a value greater than the current value may be subtracted, resulting in an (incorrect) negative value. Some of these errors may be detected, because the application assumes that these values fall within some range, so that any deviation from this range will signify an error. The wrong information is ignored in this case.
4. Link breaks between neighbors that are not mapped in the graphical model as a parent and its descendant, do not affect the message-passing process. When the link break affects the graphical model, a node may either select a new parent and try to synchronize with it, or it may become a root.

**5.3 Scalability**

The BP algorithm is not inherently localized and requires global processing of all the groups of nodes in order to achieve a global optimum. Construction of

a fully localized algorithm is similar to the general scheme with a few salient differences: (1) The localized algorithm operates locally, and therefore tends to create multiple trees, instead of a single global tree. (2) Flooding control is managed by a “propagation limit” field in each message, which determines the diameter of the message-passing tree. This field may be set to any desired small value, so only nodes within this vicinity are able to participate in the message-passing process. (3) Scalability is also achieved by defining, a priori, the number of rounds until termination, resulting in a constant message and time overhead, regardless of the size of the network. This limit is necessary not only for reducing the processing and the communication overhead, but also to ensure the termination of the process. This is due to the fact that convergence is not guaranteed in an asynchronous environment with failures and errors.

## 6 Empirical Evaluation

### 6.1 Case Study: Clustering

In this section, we analyze the BP framework that was constructed above, by applying it to an implementation of the clustering problem. The implementation provides us with a way to confirm the quality of the inference in the constructed framework.

The only aspect of this application which is implementation-specific is the content of the BP packets, and not the construction of the trees for the message-passing. Therefore we can derive conclusions from our analysis which are also applicable to other applications.

We model the sensor network as a directed graph  $G = (V, E)$ , where  $V$  is a set of nodes, where each node is assigned a local unique identifier.  $E$  is a set of wireless links connecting two adjacent nodes. Nodes are defined as adjacent if and only if they are within transmission range of each other.

The key challenge that we address here regarding clustering schemes in multi-hop WSNs is how to efficiently form a connected disjointed group of nodes in a local and distributed manner. Each group contains a single leader and several ordinary nodes. An efficient scheme is used to select cluster heads (CHs) that: (1) minimize the total transmission power aggregated over all nodes in the selected path; (2) balance the load among the nodes to prolong the network lifetime.

Optimal cluster selection is equivalent to the minimum dominating set problem, which is an NP-complete problem. Using the BP method, it is possible to achieve a good approximation in polynomial time, since the computation is dispersed and divided among all the nodes,

Following the graphical model definitions and the cost functions as presented in [9], the clustering problem can be formulated as follows:  $x_i$  is defined to be a CH candidate of node  $i$  and  $\psi_i(x_i)$  defines a local cost function of connecting node  $i$  to  $x_i$ .  $\psi_{ij}(x_i, x_j)$  represents the constraints between two neighbors  $i$  and  $j$  to prevent improper assignment of CH association. The constraints are: (1) two neighbors cannot both be CHs; (2) a node can select another node to be its CH only if that node announces that it is a CH. If one of the constraints is true, the function approaches infinity; otherwise the function approaches zero.

Broadcasting a message as part of the Min-Sum algorithm incurs a cost. Accordingly, each message stores cost information of two types: (1) The individual cost of a node to become a cluster head, independent of the other nodes. (2) The cost of connecting to other nodes, which is a function of the link between the nodes as well as other information. The information regarding these costs is updated based on the information received from the parent and the descendants in the tree structure, according to the Min-Sum algorithm. The final goal of the nodes is to select CHs that minimize the overall cost, over the whole network, based on the cost values and the constraints between the nodes [9]. In this application, two connected nodes in the graph are considered to be in the same clique if they have some predefined number of common neighbors.

## 6.2 Simulation Framework

TOSSIM, TinyOS simulator [19], was used for performance analysis of the clustering algorithm. The simulator provides an environment which is close to reality and includes realistic properties of a network, such as interferences, asymmetric links, changes in link quality, node death, failure, etc. Link Estimation and Parent Selection [20] was used as the routing protocol in the multihop network.

Every plot was taken as an average of 20 different runs and over five different time slots, to verify the behavior in different topologies of the network. The duration between two consecutive time slots was large enough for topology changes to take place and for different routing trees to be constructed, but obviously the changes are not too radical, reflecting a common trait of WSNs.

In all the simulations, the localized algorithm operates in a vicinity of two hops, with the constant number of rounds equal to eight. This number of rounds was set to guarantee convergence in an asynchronous environment when taking into account the fact that the number of rounds necessary until convergence of the clustering application is larger than the size of the tree diameter. This is because the decision about the CH candidates is done after observing the nodes in the subtree, which in itself takes several rounds. Two nodes are considered to be in the same clique when at least half of their neighbors are common.

## 6.3 Simulation Results

We determine the quality of the inference scheme by examining the number of clusters that are constructed in the network. The optimal number of clusters in WSNs depends upon network dynamics - such as connectivity and density - which change over time. Using the setup of TOSSIM (which supports at most 16 neighbors and a density of around 14 in practice), we conclude that the optimal number of clusters is about four clusters for each group of 50 nodes.

Figure 3 shows the number of CHs achieved by the localized scheme in networks with 50 to 250 nodes. As shown, the algorithm's approximation is very close to the optimal solution, which is evidence of the ability of the constructed scheme to achieve a good approximation in an inference problem.

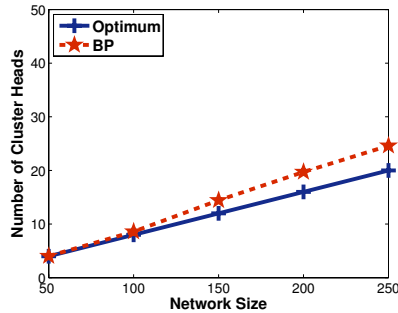


Fig. 3. Number of CHs

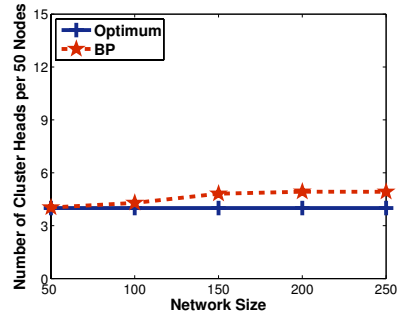


Fig. 4. CHs per 50 nodes

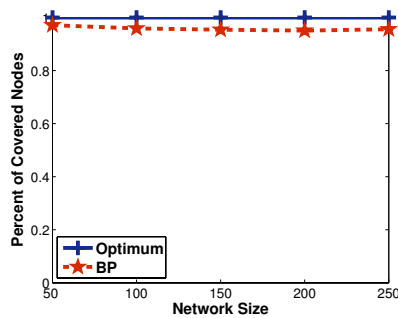


Fig. 5. Covered nodes

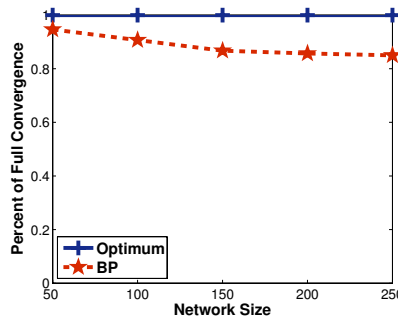


Fig. 6. Convergence percent

While it is hard to conclude the scalability of the solution from Figure 3, Figure 4 shows that the algorithm reaches a constant competitive ratio of about five cluster heads for each group of 50 nodes - as opposed to four, in medium to large groups. We chose to use a simulator that fully simulates the sensor behavior, instead of using a rough estimate given by the Matlab simulator, whose scalability constraints do not allow us to feasibly simulate networks of more than 250 sensors. However, the competitive ratio remains constant as we approach the 250-node limit. This implies that the quality of the solution will remain constant even for larger networks. The conclusion that the solution is scalable is further supported by the localized properties of the scheme, which operates in a constant vicinity with a constant number of rounds.

The convergence of the nodes into a common value, as shown in Figure 6, is inversely proportional to the number of clusters that were selected in the network. Figure 6 presents the high convergence of the algorithm, which varies between 95 to 85 percent using eight rounds. Figure 8 and Figure 7 demonstrate the average number of lost messages in the network and for each message-passing tree. It appears that large networks suffer from more errors than small networks, due to the fact that more packets are routed in these networks. These errors, in turn, slow down the convergence. When simulating networks with 200 nodes over

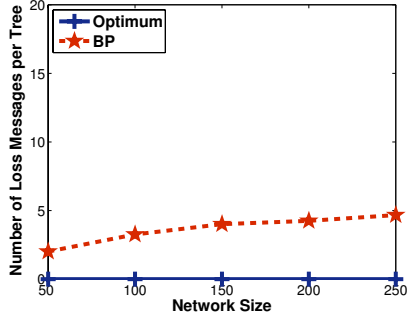


Fig. 7. Loss messages per tree

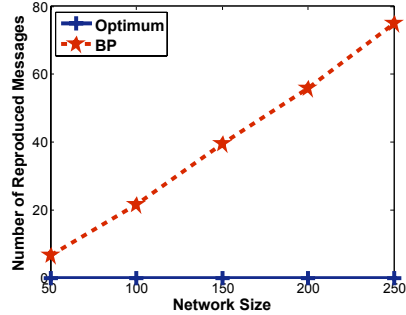


Fig. 8. Total loss messages

ten rounds instead of eight, the convergence increases from around 85 percent to about 90 percent.

To verify the effect of dynamic topology, we simulated, in networks with 250 nodes, more rapid changes in the link quality than the usual, resulting in an average of 20 significant changes during the message-passing. As can be concluded from Figure 6 and Figure 7, the scheme is robust, even in adverse network conditions, such as variable link quality.

Even in the absence of a perfect convergence, once the decision about the CHs is made, the majority of the nodes find a suitable CH in the area, as shown Figure 5. This means that the quality of the inference is very high because the clusters are spread over all the networks, and even if some rounds are missed, thus delaying the convergence at some nodes, the general inference nevertheless succeeds.

This success is due to the general inference of a value which is close to the convergence point, even in the absence of complete convergence of the nodes. The full convergence can be deduced within a few more rounds. Therefore, the final belief is almost perfect, in that it is close to that of nodes which have already converged.

## 7 Summary

This paper introduces and analyzes an efficient and practical BP framework for distributed inference. Using localized and data-centric approaches, our scheme takes into account the special characteristics and constraints of the WSN environment and consequently it provides better scaling and robustness than other approaches. With a small constant overhead, the scheme achieves outstanding results when compared to the optimal solution. The construction of the scheme is application-independent, and therefore suitable for a large variety of applications that require inference.

**Acknowledgment.** We would like to thank Danny Bickson and Amy Fredj for many valuable comments.

## References

1. Jordan, M.: Learning in Graphical Models. Kluwer Academic Publishers, Dordrecht (1998)
2. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco (1988)
3. Crick, C., Pfeffer, A.: Loopy belief propagation as a basis for communication in sensor networks. In: UAI (2003)
4. Ihler, A.T., Fisher III, J.W., Moses, R.L., Willsky, A.S.: Nonparametric belief propagation for self-calibration in sensor networks. *IEEE Journal of Selected Areas in Communication* (2005)
5. Paskin, M.A., Guestrin, C., McFadden, J.: A robust architecture for distributed inference in sensor networks. In: IPSN (2005)
6. Schiff, J., Antonelli, D., Dimakis, A.G., Chu, D., Wainwright, M.J.: Robust message-passing for statistical inference in sensor networks. In: IPSN (2007)
7. Estrin, D., Govindan, R., Heidemann, J., Kumar, S.: Next century challenges: scalable coordination in sensor networks. In: *MobiCom*. (1999)
8. Krishnamachari, B., Estrin, D., Wicker, S.: Modelling Data-Centric Routing in Wireless Sensor Networks. *IEEE INFOCOM* (2002)
9. Anker, T., Bickson, D., Dolev, D., Hod, B.: Efficient clustering for improving network performance in wireless sensor networks. In: Verdone, R. (ed.) *EWSN 2008*. LNCS, vol. 4913. Springer, Heidelberg (2008)
10. Cetin, M., Chen, L., Fisher, J., Ihler, A., Moses, R., Wainwright, M., Willsky, A.: Distributed fusion in sensor networks: A graphical models perspective. *IEEE Signal Processing Magazine* 23(4) (2006)
11. Murphy, K.P., Weiss, Y., Jordan, M.I.: Loopy belief propagation for approximate inference: An empirical study. In: UAI (1999)
12. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Understanding belief propagation and its generalizations. Technical Report TR-2001-22, MERL (2002)
13. Jordan, M.I., Weiss, Y.: Probabilistic inference in graphical models. In: *The Handbook of Brain Theory and Neural Networks*. MIT Press, Cambridge (2002)
14. Wiberg, N.: Codes and Decoding on General Graphs. PhD thesis, Dept. of Electrical Engineering, Linköping, Sweden (1996)
15. Weiss, Y.: Correctness of local probability propagation in graphical models with loops. *Neural Comput.* 12(1) (2000)
16. Chiang, M., Bambos, N.: Distributed network control through sum product algorithm on graphs. In: *GLOBECOM* (2002)
17. Prehofer, C., Bettstetter, C.: Self-organization in communication networks: principles and design paradigms. *IEEE Communications Magazine* 43(7) (2005)
18. Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed diffusion: A scalable and robust communication paradigm for sensor networks. *MobiCom* (2000)
19. TinyOs: <http://www.tinyos.net/>
20. Woo, A., Tong, T., Culler, D.: Taming the underlying challenges of reliable multi-hop routing in sensor networks. In: *SenSys*. (2003)