Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

DOAMI: A distributed on-line algorithm to minimize interference for routing in wireless sensor networks

Kejia Zhang [a], Qilong Han [a,*], Zhipeng Cai [a,b,1], Guisheng Yin [a], Junyu Lin [a]

[a] College of Computer Science and Technology, Harbin Engineering University, Harbin, 15001, China
[b] Department of Computer Science, Georgia State University, GA, USA

ARTICLE INFO

Article history:
Received 13 March 2016
Received in revised form 24 April 2016
Accepted 2 May 2016
Available online xxxx

Keywords:
Interference metric
Minimizing interference
On-line algorithm
Wireless sensor networks

ABSTRACT

This paper investigates the Minimizing-Interference-for-Multiple-Paths (MIMP) problem for routing with minimum interference in wireless sensor networks, which is defined as: Given k routing requests $\{s_1, t_1\}, \ldots, \{s_k, t_k\}$, find k paths connecting these routing requests with minimum (wireless) interference. A key point to solve this problem is how to quantify interference level among multiple routing paths. However, all the existing metrics cannot measure interference precisely under some certain circumstances, e.g., some of the routing paths have common nodes or links. This paper proposes a metric that can measure interference precisely among any set of paths, even though these paths have some nodes or links in common. For this proposed metric, the MIMP problem is NP-hard even for $k = 2$. A heuristic Distributed On-line Algorithm for Minimizing Interference (DOAMI) is given to solve the MIMP problem. DOAMI has good communication and time complexity in theory, whose efficiency is also verified by simulation results.

1. Introduction

Wireless Sensor Networks (WSNs) are more and more widely applied in our lives. More and more sensor nodes are deployed and organized as networks to monitor our homes, our work places, the environment around us and some other places we cannot reach. Since sensor nodes have only simple functional modules and limited energy supply, data exchange between two nodes usually need to be forwarded by many relaying nodes. Therefore, how to choose path for forwarding data, i.e., routing, becomes a crucial problem in WSNs. Given a routing request, i.e., two sensor nodes s and t want to communicate with each other, the main objective of routing is to optimally build paths connecting s and t (i.e., $s \sim t$ paths) for data forwarding.

The scale of a WSN can reach thousands of sensor nodes, in which data exchange could be very frequent. In such large-scale WSNs, there are usually more than one routing requests need to be met at the same time. Even if there is only one routing request, say s and t, in the network, sometimes the users may want to use multiple $s \sim t$ paths for routing. By using these paths alternately, the network can gain a better load balance. And a better throughput or reliability can be reached by using these paths for forwarding simultaneously. In such cases, we can see the multi-path routing request as k individual routing requests with the same s and t.

* Corresponding author.
E-mail addresses: hanqilong@hrbeu.edu.cn (Q. Han), zcai@gsu.edu (Z. Cai).
[1] Tel.: +1 (404) 413 5721.

http://dx.doi.org/10.1016/j.tcs.2016.05.043
0304-3975/

Please cite this article in press as: K. Zhang et al., DOAMI: A distributed on-line algorithm to minimize interference for routing in wireless sensor networks, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.05.043

To get a better routing performance, network designers always want the paths used for routing to be as independent as possible. Here, "independent" means that these routing paths do not interfere with each other. There are many works [15,16,12,19,20,1,11,14,9,7,18,17,21,10,8,23,4,2,5,3,22] study how to construct multiple independent paths in WSNs and other wireless ad hoc networks. Some works [10,8,23] borrow the thoughts from wired networks and use node-disjoint paths as independent paths. A set of paths are node-disjoint if they do not have any nodes in common. Since sensor nodes share the same communication channel in most of WSNs, when node v is transmitting data to node u, the transmission will block all v's neighbors except u from receiving data. This feature of WSNs is called *wireless interference*. Node-disjoint paths are not independent enough because of wireless interference. It has been shown in [21] that the wireless interference among multiple paths will dramatically decrease the throughput of routing even though the routing paths are node-disjoint. And [19] claims that for two node-disjoint paths, the more wireless interference between them, the larger average end-to-end delay for both paths. Therefore, the concept of non-interference paths is proposed by [17]. A set of paths are non-interference if the nodes in one path do not interfered by (are not neighbors of) any node in the other paths.

Although using non-interference paths can greatly improve the performance of routing, sometimes, especially when there are many routing requests have to be satisfied at the same time, it is impossible to find non-interference paths to satisfy all the routing requests. In such cases, the best way is relaxing the requirement for non-interference and using a set of paths with the lowest level of interference. Therefore, we need a metric to quantify interference level among multiple routing paths and an approach to build routing paths with minimum interference. There already exist some interference level metrics proposed by [15,16,12,19,20,1], but they all have apparent drawbacks. Some of them can only quantify interference level between two paths. Some of them require all the routing paths are node-disjoint, which is impossible in some cases.

In this paper, we consider the Minimizing-Interference-for-Multiple-Paths (MIMP) problem, which is defined as: Given k routing requests $\{s_1, t_1\}, \ldots, \{s_k, t_k\}$, find k paths connecting these routing requests with minimum interference. We deeply study how the transmission in different routing paths will interfere one another, especially when some of these paths share common nodes. We give a metric to measure the interference level of multiple paths. Compared with the existing interference level metrics, our metric can precisely measure the interference level among any set of routing paths, even though some of the paths share common nodes or links. Base on the proposed metric, we design a distributed on-line algorithm DOAMI (short for Distributed On-line Algorithm for Minimizing Interference) to find multiple paths with minimum interference. Therefore, the main contributions of this paper are:

- Proposing a metric that can precisely measure the interference level among any set of routing paths.
- Designing a distributed on-line algorithm DOAMI for the MIMP problem based on the proposed metric.

The rest of the paper is organized as follows. Section 2 introduces the related works; Section 3 gives the metric to measure interference level and analyzes the proposed metric; Section 4 describes the proposed algorithm DOAMI; After Section 5 verifies the efficiency of the proposed algorithm, Section 6 concludes the whole paper.

2. Related works

[15] proposes an interference-aware multi-path routing protocol IM2PR for event reporting. Upon the detection of an interesting event, the source node uses multiple node-disjoint paths to report data to the sink. These paths are constructed one by one. During the path construction, each node chooses next hop with higher delivery probability, lower interference level and more remaining energy. The interference level of node v is measured by how many neighbors of v are in the other active paths. Unfortunately, this paper does not give a clear metric to measure the interference level among multiple paths. Furthermore, IM2PR requires these multiple paths connecting the source node and the sink are node-disjoint, which is a too strong requirement for many cases.

[16] assumes that the interference range of each node is twice the transmission range and uses the conflict graph proposed by [12] to measure interference level of multiple paths. It proposes protocol I2MR to construct three non-interference paths (two for data transmission and one for backup) connecting the given source and destination. I2MR builds the shortest path from source to destination at first, then marks all the one-hop and two-hop neighbors of the first path as in the interference zone so that they can not participate in other paths. Next, I2MR finds two paths along each side of the interference zone. I2MR is only suitable for constructing interference-free paths. In many cases, it is impossible to find interference-free paths to meet all the routing requests. Besides, I2MR has an assumption which may be not true in practice, i.e., two nodes are in the twice of the transmission range from each other have at least one common neighbor.

[12] proposes the idea of using conflict graph to measure interference of multiple paths. In conflict graph H, vertices correspond to directed links in the given network. There is an edge between vertices $v_{<i,j>}$ and $v_{<p,q>}$ if links $<i, j>$ and $<p, q>$ may not be active simultaneously, i.e., node q is in the interference range of node i or node j is in the interference range of node p or $<i, j>$ and $<p, q>$ have a node in common. Based on conflict graph, this paper gives lower and upper bounds for the throughput of given source and destination node pairs. The conflict graph proposed by this work does not consider the situation that two or more routing paths have some links in common, which is inevitable if we try to route some certain node pairs simultaneously.

[19] uses the criteria called *correlation factor* to measure the interference level of multiple routing paths. The correlation factor of two node-disjoint paths is defined as the number of the links connecting the two paths. The total correlation factor

Please cite this article in press as: K. Zhang et al., DOAMI: A distributed on-line algorithm to minimize interference for routing in wireless sensor networks, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.05.043

of a set of multiple paths is defined as the sum of the correlation factor of each pair of the paths. This correlation factor criteria does not consider the situation that some routing paths have nodes or links in common either.

For a set of node-disjoint routing paths connecting a given node pair, [1] gives a metric to measure how badly one path is interfered by the other paths. For each node v_i in a routing path P, $INL(v_i)$ is defined as the number of v_i's neighbors belonging to the other routing paths. $INT(P)$ is the sum of P's intermediate nodes' INL, which measures how badly P is interfered by the other routing paths. This measurement does not consider the situation that some routing paths have common nodes either.

[11] states that even though some paths are not non-interference, we can still overcome these interference by appropriate scheduling. The key idea of the non-interference scheduling is to make the nodes on the two ends of an interference edge (inter-path edge) receive and transmit their packets simultaneously. To make the schedule works, all the nodes in the network have to be strictly synchronized.

Given a receiver r and a set of senders $S = \{s_1, s_2, \ldots, s_n\}$, [9] studies the problem of finding multiple non-interference or interference-minimized paths from S to r. Two links e_i and e_j interfere each other if the distance between an end of e_i and an end of e_j is no more than the interfering range I. For a link e in a path P, the interference of e is defined as the number of the other paths which have links interfering e. The interference level of a set of paths is defined as the maximum interference of any links in any paths. This interference metric has some apparent drawbacks. For example, paths P_1 and P_2 has only one pair of links interfering each other. The interference between P_1 and P_2 is counted as 1. Paths P_1 and P_3 has m pairs of links interfering each other. The interference between P_1 and P_3 is also counted as 1. As a result, under this metric, P_1 and P_2 have the same interference level as P_1 and P_3.

[20] defines interference level of two node-disjoint paths under signal-to-interference model. [14] also gives a metric to measure the interference level between two paths. These two works are only suitable for two paths situation. Moreover, the metric in [20] does not consider the situation that the two paths share common nodes and the metric in [14] is not symmetrical, i.e., for two paths P_1 and P_2, the interference for P_1 w.r.t. P_2 is not equal to the interference for P_2 w.r.t. P_1.

[18,17] defines that two node-disjoint paths are collision-free (non-interference) if there is no link between any two nodes on different paths. They gives algorithms to find two collision-free paths connecting a given source and destination node pair. [7] studies the flow maximization problem for given source and destination node pairs in a coordinated network while consider interference among neighboring nodes.

So far, all the existing metrics to measure interference level among multiple paths can not deal with the situation that some paths have nodes or links in common. When there are many routing requests have to be satisfied at the same time or for some certain routing requests, it is inevitable that we meet these routing requests with paths having common nodes or links. Moreover, even if we can use a set of node-disjoint paths to satisfy all the routing request, wireless interference may make the routing performance of these paths very poor. In such case, routing with certain set of paths sharing common nodes may be a much better solution.

3. Interference metric

In this section, we give our metric to measure interference level among a set of routing paths. This metric can deal with the situation that some paths have common nodes or links very well.

3.1. The scope of interference

Some of the existing works [16,9] assume that all the sensor nodes in the given network have a uniform transmission range T and a uniform interference range I, where $T \leq I \leq 2T$. Usually, I is set as $2T$. Let $d(u, v)$ denote the distance between node u and node v. There is a bidirectional wireless link (u, v) if $d(u, v) \leq T$. When node v sends data to node u, except u, the transmission will block all the nodes whose distance to v is no more than I from receiving. Such model have some important defects. First of all, the assumption that all the nodes have uniform transmission range and interference range is too ideal. For each node v, the area in which other nodes can successfully receive v's signal is affected by v's surrounding terrain. In the environments with obstacles such as mountains and indoors, for some nodes in the network, it is very likely that their signal receiving areas are not uniform circles. Second, the assumption that $I = 2T$ makes the interference level hard to be measured in practice. Now suppose the monitoring area of the given WSN is very ideal and all the nodes have uniform transmission range and interference range. When a node v is sending data, we know that the nodes in its transmission range will definitely receive v's signal, so they can not receive from other nodes at the same time. Suppose that $d(w, v) \in (T, 2T]$. From the current studies, we know that the nodes' interference range satisfies $T \leq I \leq 2T$, so w may be or may be not interfered by v's transmission. Even though w is interfered by v's transmission, this interference can not be detected or reported by w. The existing works [16,9] assume that the nodes interfered by node v's transmission must be v's one-hop or two-hop neighbor. If v decides to join into a path, it will initial a message flooding for two hops, so all the nodes receiving the message will be interfered by v's transmission. This assumption requires that if $d(w, v) \in (T, 2T]$, v and w must have at least one neighbor in common, which may be not true in a relative sparse network.

When node v is transmitting, the nodes will definitely interfered by this transmission are those nodes who can receive v's signal, i.e., the nodes who have wireless links to v. Therefore, we assume that only the nodes who have wireless links to v will be interfered by v's transmission. This assumption is more reasonable compared to the assumption in [16,9] since

Please cite this article in press as: K. Zhang et al., DOAMI: A distributed on-line algorithm to minimize interference for routing in wireless sensor networks, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.05.043

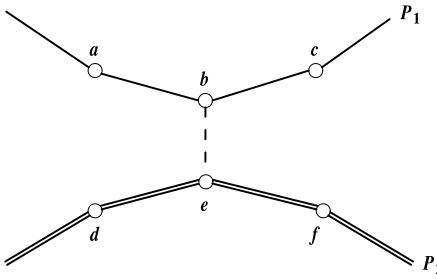


Fig. 1. Example 1 with two node-disjoint paths (single lines denote edges in P_1, double lines denote edges in P_2, dashed lines denote cross-path edges).

it only count the interference which will surely happen and it does not require the signal receiving area of each node to be a uniform circle. Furthermore, our assumption makes interference level easy to be measured, since the nodes which will be interfered by v's transmission must be v's (one-hop) neighbor.[2]

3.2. Network model

We model the given network as an undirected graph $G = (V, E)$, where $V = \{v \mid v \text{ is a sensor node}\}$ and $E = \{(u, v) \mid$ there is a wireless link between $u \in V$ and $v \in V\}$. $n = |V|$ is the number of nodes and $m = |E|$ is the number of wireless links in the given network. Here, we assume that all the links are bidirectional, *i.e.*, if v can receive the signal from u then u can receive the signal from v. $N(v)$ is the set including all v's neighbors. Hereinafter, we do not distinguish the terms "network" and "graph" and the terms "edge" and "wireless link".

A path P in G is a subgraph which can be expressed as a sequence of distinct nodes $(v_1, \ldots, v_m)$, where $v_i \in V$ for $i = 1, \ldots, m$ and $(v_i, v_{i+1}) \in E$ for $i = 1, \ldots, m-1$. Path $P = (v_1, \ldots, v_m)$ is a $v_1 \sim v_m$ path. We say a path P is the *acyclic path* if there is no edge connecting two non-adjacent nodes in P. Since the paths we construct are all the shortest paths w.r.t. some criteria, the term "path" means acyclic path in the rest of this paper.

Although we model the given network as an undirected graph, we still emphasize the direction of a path by nodes' order in its expression or the words "from" and "to". Thus, the direction of path $P = (v_1, \ldots, v_{i-1}, v_i, v_{i+1}, \ldots, v_m)$ is from v_1 to v_m. v_i's previous hop in P is v_{i-1} and its next hop in P is v_{i+1}. For each path P_i connecting a given routing request $\{s_i, t_i\}$, we set its default direction is from s_i to t_i.

Suppose that k routing paths $P_1, \ldots, P_k$ are being used in the given network. Next, we discuss how to measure interference level among these paths. We say an edge is a *cross-path edge* if its two end nodes belong to different paths of $P_1, \ldots, P_k$. We use the function $C : V \to \mathbb{N}^0$ to define how many paths each node is in.

3.3. Quantifying interference level

By our assumption of nodes' interference scope, if $P_1, \ldots, P_k$ are node-disjoint, it is reasonable to use the number of cross-path edges to measure their interference level as was done in [19]. In Example 1 with two node-disjoint paths as shown in Fig. 1, the cross-path edge (b, e) corresponds to the interference between node b in P_1 and node e in P_2. When b is sending messages for P_1, e cannot receive messages in P_2. When e is sending messages for P_2, b cannot receive messages in P_1. Therefore, for a set of node-disjoint paths, each cross-path edge corresponds to such a two-way send-and-receive interference between two nodes in different paths.

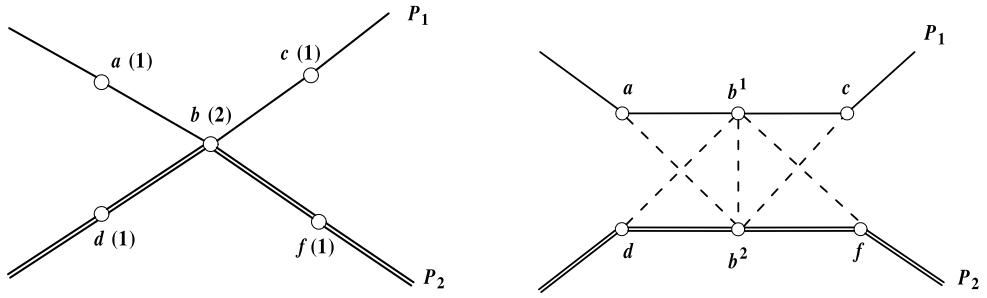
As we mentioned before, in many cases, it is impossible to find a set of node-disjoint path to meet all the routing requests. Sometimes, because of wireless interference, routing with paths sharing common nodes may be a better solution than routing with node-disjoint paths. Therefore, the proposed interference metric must be able to deal with the situation that some of the given routing paths have nodes or edges in common.

Let us consider how interference will happen in Example 2 with two paths sharing a common node as shown in Fig. 2(a). P_1 and P_2 have a common node b. The edge (b, d) can be seen as a cross-path edge which corresponds to the two-way send-and-receive interference between b in P_1 and d in P_2. Similarly, the edges (b, f), (b, a) and (b, c) can be seen as cross-path edges too. When b is sending messages for P_1, it cannot receive messages in P_2. When b is sending messages for P_2, it cannot receive messages in P_1 either. Thus, there is a two-way send-and-receive interference between b in P_1 and b in P_2. To get an equivalent case with two node-disjoint paths, we can split b into two nodes b^1 and b^2 as shown in Fig. 2(b). b^1 is the b in P_1 and b^2 is the b in P_2. Since there is a two-way send-and-receive interference between b in P_1 and b in P_2, we add a cross-path edge (b^1, b^2). From this equivalent case, we can easily see that the interference level between P_1 and P_2 (the number of cross-path edges) is 5.

We extend the above analysis to measure interference level among k paths $P_1, \ldots, P_k$. Suppose u and v are neighbors of each other, $C(u) > 1$ and $C(v) > 1$. To get the node-disjoint equivalent case of $P_1, \ldots, P_k$, we split u into $C(u)$ new

[2] Hereinafter, if not otherwise specified, "neighbor" means one-hop neighbor.

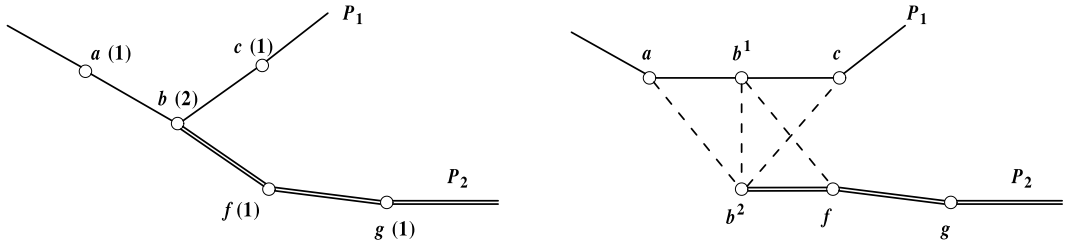
Please cite this article in press as: K. Zhang et al., DOAMI: A distributed on-line algorithm to minimize interference for routing in wireless sensor networks, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.05.043



(a) Example 2 (two paths sharing a common node)

(b) The node-disjoint equivalent case of Example 2

Fig. 2. Example 2 and its node-disjoint equivalent case (single lines denote edges in P_1, double lines denote edges in P_2, dashed lines denote cross-path edges, numbers in the brackets denote nodes' $C(v)$).



(a) Example 3 (P_2 sharing its end node with P_1)

(b) The node-disjoint equivalent case of Example 3

Fig. 3. Example 3 and its node-disjoint equivalent case (single lines denote edges in P_1, double lines denote edges in P_2, dashed lines denote cross-path edges, numbers in the brackets denote nodes' $C(v)$).

nodes $u^1, \ldots, u^{C(u)}$ and split v into $C(v)$ new nodes $v^1, \ldots, v^{C(v)}$. There is an edge connecting each u^i $(i = 1, \ldots, C(u))$ and each v^j $(j = 1, \ldots, C(v))$. All these $u^1, \ldots, u^{C(u)}$ form a clique, so do $v^1, \ldots, v^{C(v)}$. Therefore, in the node-disjoint equivalent case, if we only count the edges connecting the nodes in $P_1, \ldots, P_k$, the number of edges at each v^j will be $C(v) - 1 + \sum_{u \in N(v)} C(u)$. In those edges, except the edges connecting v^j to its previous hop and next hop in the same path, all the other edges are cross-path edges. Therefore, the number of cross-path edges at each v^j is

$$W(v) - 3$$

where $W : V \to \mathbb{N}^0$ is defined as $W(v) = C(v) + \sum_{u \in N(v)} C(u)$. The total number of cross-path edges at all these $v^1, \ldots, v^{C(v)}$ is

$$C(v)\big(W(v) - 3\big)$$

The total number of cross-path edges in the node-disjoint equivalent case of $P_1, \ldots, P_k$, i.e., the interference level among these paths is

$$\frac{\sum_{v \in V} C(v)\big(W(v) - 3\big)}{2}$$

In Example 2. We have $W(b) = 6, W(a) = W(c) = W(e) = W(f) = 4$ (although the other neighbor of a is not shown in the figure, we suppose that its $C(v)$ is 1). The number of cross-path edges at b^1 / b^2 in the node-disjoint equivalent case is $W(b) - 3 = 3$. The number of cross-path edges at $a/b/e/f$ is $4 - 3 = 1$. The total number of cross-path edges in the node-disjoint equivalent case is $(2 * 3 + 1 + 1 + 1 + 1)/2 = 5$.

The above analysis ignore the particularity of the two ends of each path. For the $s_i \sim t_i$ path P_i, its starting node s_i does not have previous hop and its ending node t_i does not have next hop. Therefore, in the node-disjoint equivalent case, the number of cross-path edges at each end of these paths should be $W(v) - 2$ rather than $W(v) - 3$. In Example 3 as shown in Fig. 3(a), b is the end of P_2 and also an intermediate node in P_1. In the node-disjoint equivalent case of Example 3, the number of cross-path edges at b^1 is $W(b) - 3 = 2$, whereas the number of cross-path edges at b^2 is $W(b) - 2 = 3$. Since there are $2k$ ends in the node-disjoint equivalent case for $P_1, \ldots, P_k$, the total number of cross-path edges can be given by Definition 3.1.

Please cite this article in press as: K. Zhang et al., DOAMI: A distributed on-line algorithm to minimize interference for routing in wireless sensor networks, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.05.043

Definition 3.1. For a set of paths $P_1, \ldots, P_k$, their interference level is defined as

$$IN(P_1, \ldots, P_k) = \frac{\sum_{v \in V} C(v)\big(W(v) - 3\big)}{2} + k$$

where $C(v)$ is how many paths v is in and $W(v) = C(v) + \sum_{u \in N(v)} C(u)$.

3.4. Analysis of interference metric

Suppose that $P_1, \ldots, P_k$ are constructed one by one in the order of their suffixes. Define $C_i(v)$ $(i = 1, \ldots, k)$ as how many paths v is in after deploying $P_1, \ldots, P_i$ and $C_0(v) = 0$ for each $v \in V$. Accordingly, $W_i(v) = C_i(v) + \sum_{u \in N(v)} C_i(u)$. Since $P_1, \ldots, P_k$ are acyclic paths, we have the following Proposition.

Proposition 1. *For each intermediate node v in P_{i+1}, $W_{i+1}(v) = W_i(v) + 3$. For each end u of P_{i+1}, $W_{i+1}(u) = W_i(u) + 2$. And*

$$\sum_{v \in P_{j+1}} W_j(v) = 2 + \sum_{v \in P_{j+1}} \big(W_{j+1}(v) - 3\big).$$

Now, let us consider the change of interference level before and after deploying path P_{i+1}. For simplicity, we use IN_i to denote $IN(P_1, \ldots, P_i)$, and use C_i and W_i to denote $C_i(v)$ and $W_i(v)$ respectively. By the definition of interference level we have

$$2(IN_{i+1} - IN_i) = 2(i+1) - 2i + \sum_{v \in V} \big(C_{i+1}(W_{i+1} - 3) - C_i(W_i - 3)\big) \tag{3.1}$$

$$= 2 + \sum_{v \in P_{i+1}} \big(C_{i+1}(W_{i+1} - 3) - C_i(W_i - 3)\big) + \sum_{v \in V - P_{i+1}} \big(C_{i+1}(W_{i+1} - 3) - C_i(W_i - 3)\big) \tag{3.2}$$

$$= 2 + \sum_{v \in P_{i+1}} \big((C_i + 1)(W_{i+1} - 3) - C_i(W_i - 3)\big) + \sum_{v \in V - P_{i+1}} \big(C_i(W_{i+1} - 3) - C_i(W_i - 3)\big) \tag{3.3}$$

$$= 2 + \sum_{v \in P_{i+1}} (W_{i+1} - 3) + \sum_{v \in P_{i+1}} C_i(W_{i+1} - W_i) + \sum_{v \in V - P_{i+1}} C_i(W_{i+1} - W_i) \tag{3.4}$$

$$= \sum_{v \in P_{i+1}} W_i + \sum_{v \in V} C_i(W_{i+1} - W_i) \tag{3.5}$$

Equation (3.3) can be get from Equation (3.2) because $C_{i+1}(v) = C_i(v) + 1$ for each $v \in P_{i+1}$ and $C_{i+1}(v) = C_i(v)$ for each $v \in V - P_{j+1}$. By Proposition 1, we can get Equation (3.5) from Equation (3.4).

By the definition of function W, we have

$$\sum_{v \in P_{i+1}} W_i(v) = \sum_{v \in P_{i+1}} C_i(v) + \sum_{v \in P_{i+1}} \sum_{u \in N(v)} C_i(u) \tag{3.6}$$

$$= \sum_{v \in P_{i+1}} C_i(v) + \sum_{\substack{v \in P_{i+1} \\ (u,v) \in E}} C_i(u) \tag{3.7}$$

and

$$\sum_{v \in V} C_i\big(W_{i+1}(v) - W_i(v)\big) = \sum_{v \in V} C_i(v)\big(C_{i+1}(v) - C_i(v)\big) + \sum_{v \in V} \Big(C_i(v) \sum_{u \in N(v)} \big(C_{i+1}(u) - C_i(u)\big)\Big) \tag{3.8}$$

$$= \sum_{v \in P_{j+1}} C_i(v) + \sum_{\substack{u \in P_{j+1} \\ (u,v) \in E}} C_i(v) \tag{3.9}$$

Please cite this article in press as: K. Zhang et al., DOAMI: A distributed on-line algorithm to minimize interference for routing in wireless sensor networks, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.05.043

We can get Equation (3.9) from Equation (3.8) because $C_{i+1}(v) - C_i(v) = 1$ for $v \in P_{j+1}$ and $C_{i+1}(v) - C_i(v) = 0$ for $v \in V - P_{j+1}$.

From Equation (3.7) and (3.9), we know

$$\sum_{v \in P_{i+1}} W_i(v) = \sum_{v \in V} C_i\big(W_{i+1}(v) - W_i(v)\big) \tag{3.10}$$

Combine Equation (3.10) and (3.5), we have Lemma 1 and Corollary 1.

Lemma 1. *For $i = 0, 1, \ldots, k-1$, the interference level among paths satisfies*

$$IN(P_1, \ldots, P_{i+1}) - IN(P_1, \ldots, P_i) = \sum_{v \in P_{i+1}} W_i(v)$$

Corollary 1. *The interference level among paths $\{P_1, \ldots, P_k\}$ can be also computed by*

$$IN(P_1, \ldots, P_k) = \sum_{v \in P_1} W_0(v) + \sum_{v \in P_2} W_1(v) + \cdots + \sum_{v \in P_k} W_{k-1}(v)$$

Corollary 1 gives us the idea of designing on-line algorithm for MIMP. We satisfy the given routing requests $\{s_1, t_1\}, \ldots, \{s_k, t_k\}$ in order. For each $\{s_i, t_i\}$, we find the shortest $s_i \sim t_i$ path w.r.t. function W_{i-1}. Here, the length of path P_i w.r.t. W_{i-1} is defined as $\sum_{v \in P_i} W_{i-1}(v)$.

4. Distributed on-line algorithm

In this paper, we consider the MIMP problem, which is defined as: Given k routing requests $\{s_1, t_1\}, \ldots, \{s_k, t_k\}$, finding k paths $P_1, \ldots, P_k$ satisfying these routing requests, i.e., P_i is a $s_i \sim t_i$ path for $i = 1, \ldots, k$. The optimizing goal is to minimize interference among these paths, i.e., $IN(P_1, \ldots, P_k)$.

According to [13], given two node pairs $\{s_1, t_1\}$ and $\{s_2, t_2\}$ in a graph, it is NP-complete to determine whether there exist a pair of non-interference (there is no cross-path edge) $s_1 \sim t_1$ path and $s_2 \sim t_2$ path. Therefore, we have Theorem 1.

Theorem 1. *MIMP is NP-hard even for $k = 2$.*

MIMP is very hard to solve or to approximate. In this paper, we design a heuristic on-line algorithm DOAMI for the MIMP problem. DOAMI is executed in a totally distributed way, i.e., each node participates by only exchanging short messages with its neighbors and the exchange of messages does not rely on any pre-defined structure. DOAMI is also an on-line algorithm, which means that DOAMI deploys paths to satisfy the given routing requests one by one. After DOAMI receives a routing request $\{s_i, t_i\}$, it immediately finds a $s_i \sim t_i$ path to satisfy it without waiting for the coming of the next routing request.

4.1. Description of DOAMI

The design of DOAMI is motivated by Corollary 1. DOAMI uses two phases to find path for each routing request $\{s_i, t_i\}$. In Phase 1, starting from s_i, by exchanging FIND-SWP messages, each node v finds the shortest path from s_i to itself w.r.t. function W_{i-1}. In Phase 2, starting from t_i, the shortest path from s_i to t_i w.r.t. W_{i-1} is traced and confirmed. Meanwhile, each node computes its $W_i(v)$. DOAMI is given by Algorithm 1. We assume that the ID of the sender of each message is contained in the message's header, so the receiver can always know who is the sender of the received message. In the following, "SWP" means shortest path w.r.t. function W_{i-1}.

Each node v maintains four variables $W(v), W\_len(v), SWP\_LP(v), SWP\_NP(v)$. While executing routing request $\{s_i, t_i\}$, $W(v)$ records $W_{i-1}(v)$ in Phase 1 and is updated to $W_i(v)$ in Phase 2. $W\_len(v)$ records the length of the SWP from s_i to v found by now. Let P_i be the SWP from s_i to t_i found by DOAMI at the end of Phase 1. If v is a node in P_i, variables $SWP\_LP(v)$ and $SWP\_NP(v)$ record v's last and next hop in P_i respectively.

In Phase 1, by exchanging FIND-SWP messages, each node v finds the SWP from s_i to itself. The data field of each FIND-SWP message (let the sender be u) contains two variables: the destination node's ID t_i and the length of the SWP from s_i to u, i.e., $W\_len(u)$. At first, the source node s_i broadcasts a FIND-SWP message $\{t_i, W\_len(s_i)\}$ to start Phase 1 (Line 1 in Algorithm 1). While a node $v \neq s_i$ receives a FIND-SWP message $\{t_i, W\_len(u)\}$ from u, it finds a new path from s_i to u then to itself, whose W_{i-1}-length is $W\_len(u) + W(v)$. If the new path is shorter (w.r.t. W_{i-1}) than the current SWP from s_i to v, it should be the new SWP from s_i to v. In this case, v updates $W\_len(v)$ to $W\_len(u) + W(v)$ and updates $SWP\_LP(v)$ to u, then broadcasts a FIND-SWP message $\{t_i, W\_len(v)\}$ (Line 2–5, 8, 9 in Algorithm 1). From receiving the first FIND-SWP message, t_i waits for the end of Phase 1 then broadcasts a TRACE-SWP message to start Phase 2 (Line 6, 7, 10 in Algorithm 1).

Please cite this article in press as: K. Zhang et al., DOAMI: A distributed on-line algorithm to minimize interference for routing in wireless sensor networks, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.05.043

Input: k routing requests $\{s_1, t_1\}, \ldots, \{s_k, t_k\}$
Output: k paths $P_1, \ldots, P_k$ that P_i is a $s_i \sim t_i$ path for $i = 1, \ldots, k$
/* codes to deal with each $\{s_i, t_i\}$ */
// Phase 1:
s_i sets $W\_len(s_i) = W(s_i)$ and broadcasts a FIND-SWP message $\{t_i, W\_len(s_i)\}$;
while $v \neq s_i$ receives a FIND-SWP message $\{t_i, W\_len(u)\}$ from u **do**
  if $W\_len(v) > W\_len(u) + W(v)$ **then**
    $W\_len(v) \leftarrow W\_len(u) + W(v)$;
    $SWP\_LP(v) \leftarrow u$;
    if $v = t_i$ **then**
      Go to Phase 2;
    else
      Broadcast a FIND-SWP message $\{t_i, W\_len(v)\}$;
    end
  end
end
// Phase 2:
t_i waits for the end of FIND-SWP message exchanging then sends a TRACE-SWP message to $SWP\_LP(t_i)$;
while v receives a TRACE-SWP message from u **do**
  $SWP\_NP(v) \leftarrow u$;
  $W(v) \leftarrow W(v) + 1$;
  Broadcast a PATH-CONF message;
  Send a TRACE-SWP message to $SWP\_LP(v)$;
end
while v receives a PATH-CONF message **do**
  $W(v) \leftarrow W(v) + 1$;
end

Algorithm 1: Distributed On-line Algorithm for Minimizing Interference (DOAMI).

Let P_i be the SWP from s_i to t_i found by DOAMI in Phase 1. In Phase 2, starting from t_i, each node in P_i sends a TRACE-SWAP message to its last hop in P_i. In this way, P_i is confirmed and established. Each TRACE-SWP message has an empty data field. In the process of confirming P_i, each node in the network should update their $W(v)$. In Phase 1, $W(v)$ records $W_{i-1}(v)$. At the end of Phase 2, $W(v)$ should be updated to $W_i(v)$. While node v receives a TRACE-SWP message from u, it sets u as its next hop in P_i (Line 12 in Algorithm 1). Since v joins into P_i, $C(v)$ should be increased by one and all the neighbors of v should add one to their W variable. Therefore, we let v add 1 to $W(v)$ (Line 13 in Algorithm 1) and broadcast a PATH-CONF message with empty data field to inform its neighbors. Receiving a PATH-CONF message, each node adds one to their W variable (Line 16, 17 in Algorithm 1).

We refer the execution time for each routing request $\{s_i, t_i\}$ as a "round". We assume that for each node v, $W\_len(v)$ is positive infinity at the beginning of each round. This can be implemented by involving a round number i in each FIND-SWP message. While v receives a FIND-SWP message $\{t_i, W\_len(u)\}$ with a larger round number, no matter what value $W\_len(v)$ is, it updates $W\_len(v)$ to $W\_len(u) + W(v)$.

4.2. Improving DOAMI

In each round of the DOAMI's execution, it is hard for t_i to judge when Phase 1 will end. Basically, t_i can do nothing but wait for a very long time. This is an apparent drawback of DOAMI. On the other hand, for a node v, if a FIND-SWP message along a longer (w.r.t. W_{i-1}) path from s_i to v always arrives at v earlier, v will frequently update its $W\_len(v)$ and broadcast FIND-SWP message. The frequently updating of v will further cause frequently updating of v's neighbors. This is another drawback of DOAMI.

To overcome the above two drawbacks, we design a delay-inform mechanism to improve DOAMI. Suppose that node v receives a FIND-SWP message and updates its $W\_len(v)$. Instead of broadcasting a FIND-SWP message including the new $W\_len(v)$ immediately, v waits for $W(v) * T_0$ time then broadcasts the FIND-SWP message. T_0 is an appropriate time unit. While waiting for the broadcasting, if v receives another FIND-SWP message and updates its $W\_len(v)$ again, v will cancel the broadcasting and set timer for another broadcasting.

This delay-inform mechanism guarantees that a FIND-SWP message along a shorter (w.r.t. W_{i-1}) path from s_i to v will arrive at v earlier. Each node v only needs to set its $W\_len(v)$ and broadcast a FIND-SWP message once while it receives the first FIND-SWP message. $W\_len(v)$ will not be updated again after that. While t_i receives its first FIND-SWP message, t_i does not wait and immediately sends a TRACE-SWP message to $SWP\_LP(t_i)$ to start Phase 2. While s_i receives a TRACE-SWP message, after it broadcasts a PATH-CONF message, the execution of the current round is done.

By applying the delay-inform mechanism, in each round, each node sends out at most one FIND-SWP message in Phase 1 and at most one TRACE-SWP message and one PATH-CONF message in Phase 2. Since all the three kinds of messages have constant length, the communication complexity of DOAMI is $O(k * n)$.

Let P_i be the shortest $s_i \sim t_i$ path w.r.t. function W_{i-1}. The execution time for Phase 1 of the ith round is $T_0 * \sum_{v \in P_i} W_{i-1}(v)$, where T_0 is the unit time. By Equation (3.7), we have

Please cite this article in press as: K. Zhang et al., DOAMI: A distributed on-line algorithm to minimize interference for routing in wireless sensor networks, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.05.043

$$\sum_{v\in P_i} W_{i-1}(v) = \sum_{v\in P_i} C_{i-1}(v) + \sum_{\substack{v\in P_i \\ (u,v)\in E}} C_{i-1}(u) \tag{4.1}$$

$$\leq \sum_{\substack{v\in P_i \\ (u,v)\in E}} \big(C_{i-1}(u) + C_{i-1}(v)\big) \tag{4.2}$$

$$\leq \sum_{(u,v)\in E} \big(C_{i-1}(u) + C_{i-1}(v)\big) \tag{4.3}$$

$$\leq 2k * m \tag{4.4}$$

Formula (4.4) can be get from Formula (4.3) because $C_{i-1}(v) \leq k$ for each v. The execution time for Phase 2 of the ith round is less than n. Therefore, the time complexity of DOAMI is $O(k^2 * m)$.

5. Simulation results

We use a simulator written in C++ to evaluate the proposed algorithm. We set the simulation environment as follows: 2500 sensor nodes are randomly deployed in a 1500 m $\times$ 1500 m area. The effective transmitting range T of each sensor node is set to 50 m. Let v be the sender of any data packet. There are two kinds of interference caused by v's transmission: one-hop interference and two-hop interference. The nodes (besides the receiver) whose distance to v is less than or equal to 50 m will detect v's signal and be interfered by the transmission for sure. On the other hand, the nodes whose distance to v is in (50 m, 100 m] has a certain probability to be interfered by v's signal. We call this probability the Interference Probability (IP). Two-hop interference cannot be detected by the interfered node. In the simulation, we divide the time into equal-sized time slots. The transmission of any data packet can be done in one time slot. Each node can only send or receive one data packet (can not both send and receive) in one time slot. If a node is interfered by some transmission in a time slot, it cannot receive any data packet in this time slot. The sender of a packet can know that the transmission is successful by receiving the receiver's hand-shake message. If the sender does not receive the hand-shake message from the receiver in the current time slot, it will retransmit the message in the next time slot.

To simulate energy consumption, we set the parameters of sensor nodes according to TelosB motes [6]. The supply voltage is 3.0 V. When MCU is "on" and Radio is on TX or RX mode, the current is 22 mA. The transmit bit rate is 250 kbps. Therefore, the energy consumption for the sender to send 1 bit data (or for the receiver to receive 1 bit data) successfully is 2.64×10^{-7} J. For the TelosB sensor node, the energy consumption to transmit one bit data can be used for MCU to execute more than 400 instructions, so we neglect the energy consumption for computing.

To simulate routing, we randomly generate different number of routing requests, i.e., $\{s, t\}$ pairs. Given $\{s_1, t_1\}, \ldots, \{s_k, t_k\}$, we use routing algorithm to built paths to satisfy all the routing requests. Next, simultaneously, for each $\{s_i, t_i\}$, we let s_i generate a new data packet in every 3 time slots and send the packets to t_i along the built path. Each s_i and t_i have to transmit 100 data packets. Each data packet has a 6-byte header containing the information of receiver's ID (2 bytes), sender's ID (2 bytes), packet type (1 bytes) and packet length (1 bytes). The data field of each packet contains 50 bytes. The efficiency of routing is measured at four aspects: 1) Transmission Delay. It is the number of time slots we use to complete the transmission of data packets for all the routing requests. This figure indicates the throughput of routing. 2) Routing Energy. It is the energy we use to transmit the data packets for all the routing request. 3) Waste Energy. It is the energy spent on retransmission caused by interference. 4) Build Path Energy. It is the energy we use to build paths for all the routing requests.

We compare the proposed algorithm DOAMI with three other algorithms:

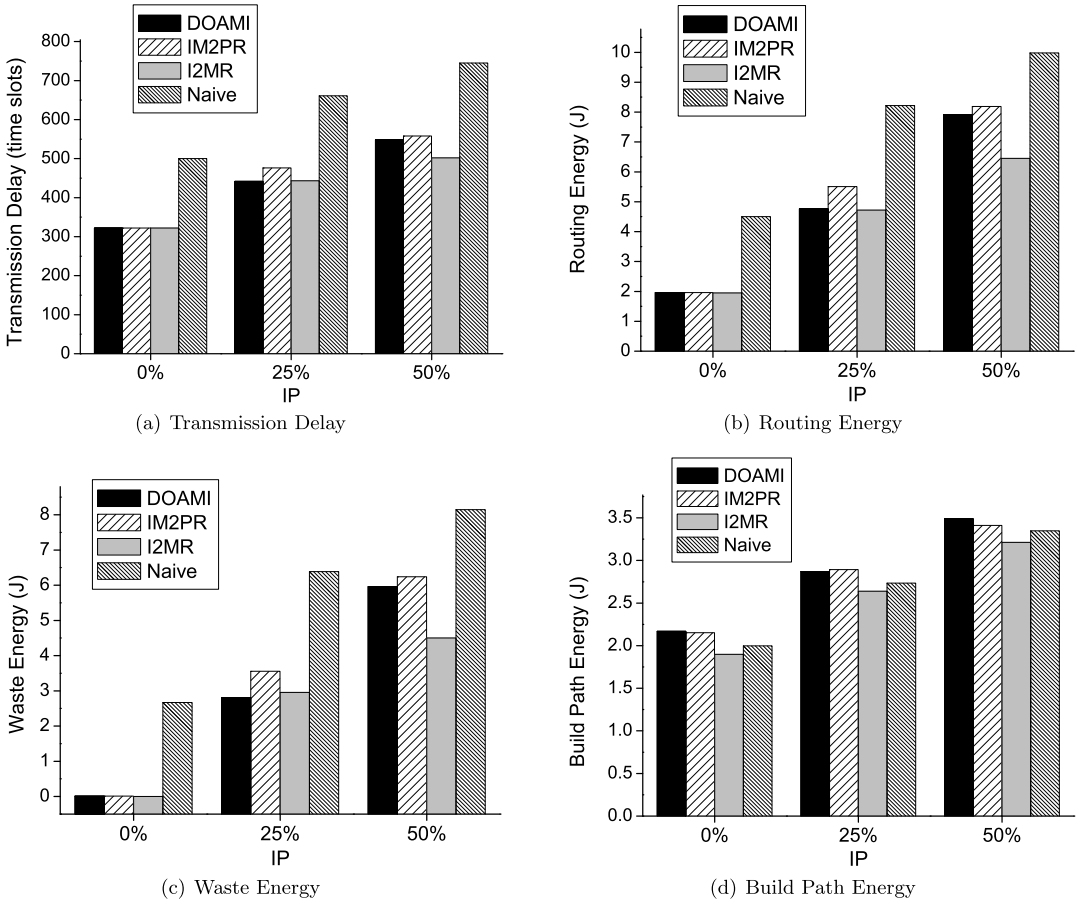
- IM2PR. It is proposed by [15]. It builds path for the routing requests one by one. For each routing request $\{s_i, t_i\}$, it builds the path from s_i to t_i with the lowest interference level. The interference level of node v is measured by how many neighbors of v are in the other active paths. IM2PR requires the built paths to be node-disjoint.
- I2MR. It is proposed by [16]. It also can be seen as an on-line algorithm. For each routing request $\{s_i, t_i\}$, it builds the shortest paths from s_i to t_i and marks all the two-hop neighbors of the path nodes "deleted" so that they will not join other paths. Therefore, any two nodes in different paths are more than two hops away from each other.
- Naive. It is the algorithm without considering interference. For each routing request $\{s_i, t_i\}$, it simply finds the shortest paths (with smallest number of hops) from s_i to t_i.

At first, we compare the largest number of routing requests each algorithm can satisfy. From Table 1, we can see that DOAMI and Naive can satisfy all the given k routing requests no matter what value k is. In contrast, IM2PR can build paths for at most 13 routing requests. Since IM2PR requires that all the built paths are node-disjoint, when k becomes larger, it is hard to find a path for the latter routing request to get through the "barrier" formed by the former built paths. I2MR gives the worst performance in this aspect because it has the strictest requirement of the built paths. The paths built by I2MR have to be more than two hops away from each other. In the following, we only compare the algorithms when they build the same number of routing paths.

Please cite this article in press as: K. Zhang et al., DOAMI: A distributed on-line algorithm to minimize interference for routing in wireless sensor networks, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.05.043

Table 1
The largest number of satisfying routing requests.

DOAMI	IM2PR	I2MR	Naive
∞	13	5	∞



(a) Transmission Delay (b) Routing Energy

(c) Waste Energy (d) Build Path Energy

Fig. 4. Simulation results of group 1: $k = 5$.

In the first group of simulation, we set the number of routing requests as $k = 5$ and compare all the four algorithms. The comparison is done at three different values of IP, i.e., in what probability the nodes in $(R, 2R]$ from sender will be interfered by the transmission. The simulation results are given in Fig. 4.

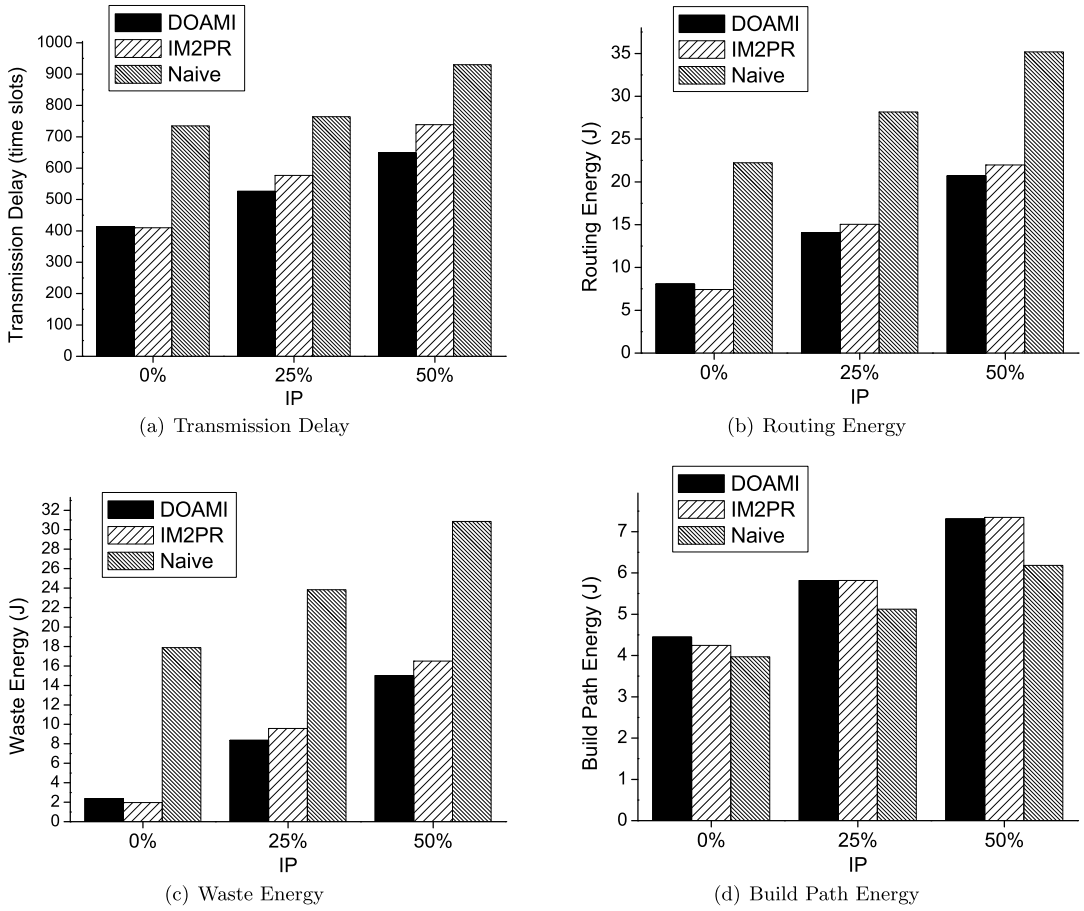
As shown in Fig. 4(a), I2MR has the best performance in the aspect of Transmission Delay. Since I2MR has the strictest requirement of the routing paths, the transmissions in different paths barely interfere with each other. DOAMI outperforms IM2PR a little in this aspect when IP becomes larger. The difference among I2MR, DOAMI and IM2PR is very limited. Since Naive does not consider the interference while building routing paths, it performs worst in this aspect.

As shown in Fig. 4(b), I2MR also has the best performance in the aspect of Routing Energy. Since I2MR almost avoids all one-hop and two-hop interference among different paths, when IP become larger, the advantage of I2MR gets more and more apparent. However, when $k > 5$, I2MR cannot satisfy all the routing requests. In this aspect, DOAMI and IM2PR almost have the same performance and DOAMI outperforms IM2PR a little. Naive still gives the worst performance. As shown in Fig. 4(c), Waste Energy can be seen as the reflection of Routing Energy since an algorithm spends less energy for routing if it has smaller number of retransmission.

As shown in Fig. 4(d), in the aspect of Build Path Energy, the difference among the four algorithms is very small. I2MR and Naive outperform DOAMI and IM2PR a little. Since I2MR and Naive are simpler, the nodes exchanges less information in the process of building paths.

In the second group of simulation, we set $k = 10$ to compare DOAMI, IM2PR and Naive. The simulation results are given in Fig. 5. As shown in Fig. 5(a), 5(b) and 5(c), DOAMI has the best performance in the aspects of Transmission Delay, Routing Energy and Waste Energy. When IP gets larger, DOAMI outperforms the other two algorithms more and more apparently. In these three aspects, Naive always gives the worst performance since it does not consider interference. As shown in Fig. 5(d),

Please cite this article in press as: K. Zhang et al., DOAMI: A distributed on-line algorithm to minimize interference for routing in wireless sensor networks, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.05.043



(a) Transmission Delay (b) Routing Energy

(c) Waste Energy (d) Build Path Energy

Fig. 5. Simulation results of group 2: $k = 10$.

since Naive is simpler, it uses less energy to build routing paths. DOAMI and IM2PR almost have the same performance in the aspect of Build Path Energy. The difference among the three algorithms in this aspect is totally tolerable.

In the third group of simulation, we set $k = 20$ to compare DOAMI and Naive. The simulation results are given in Fig. 6. As shown in Fig. 6(a), 6(b) and 6(c), DOAMI apparently outperforms Naive in the aspects of Transmission Delay, Routing Energy and Waste Energy. Although Naive uses less energy to build paths as shown in Fig. 6(d), DOAMI is still much better by combining Routing Energy and Build Path Energy together.

Let us summary the simulation results. For both routing throughput and energy efficiency, I2MR is a little better than DOAMI when k is small. However, I2MR cannot deal with the cases with a relative large k. DOAMI is a little better than IM2PR and much better than Naive in both aspects of routing throughput and energy efficiency. Compared with IM2PR, DOAMI has the ability to deal with larger k whereas IM2PR cannot.

6. Conclusion

In this paper, we give a metric to measure interference level among multiple routing paths in wireless sensor networks. Unlike the existing metrics, the proposed metric can deal with the situation that the routing paths have some nodes or links in common. Based on the metric, we propose a distributed on-line algorithm DOAMI to build paths with minimum interference for multiple routing requests. The efficiency of DOAMI is confirmed by our simulation results. For the future works, maybe we will consider how to design an algorithm for the same problem with guaranteed bound.

Acknowledgement

This work is supported by National Natural Science Foundation of China (Grant No. 61300207, Grant No. 61272186, Grant No. 61370084), Fundamental Research Funds for the Central Universities (Grant No. HEUCF160604).

Please cite this article in press as: K. Zhang et al., DOAMI: A distributed on-line algorithm to minimize interference for routing in wireless sensor networks, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.05.043

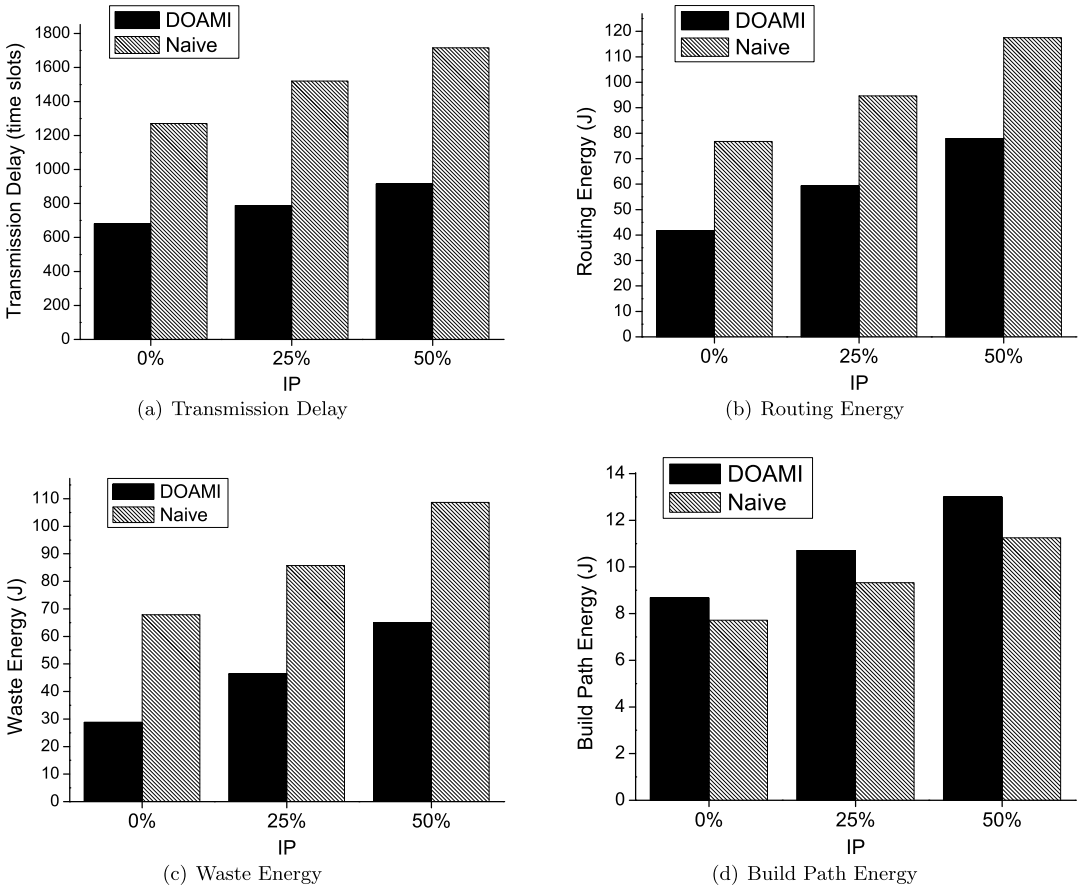


Fig. 6. Simulation results of group 3: $k = 20$.

References

[1] Zahia Bidai, Moufida Maimour, Interference-aware multipath routing protocol for video transmission over ZigBee wireless sensor networks, in: International Conference on Multimedia Computing and Systems, ICMCS, 2014, pp. 837–842.
[2] Zhipeng Cai, Zhi-Zhong Chen, Guohui Lin, A 3.4713-approximation algorithm for the capacitated multicast tree routing problem, Theoret. Comput. Sci. 410 (52) (2008) 5415–5424.
[3] Zhipeng Cai, Zhizhong Chen, Guohui Lin, An improved approximation algorithm for the capacitated multicast tree routing problem, in: The 2nd Annual International Conference on Combinatorial Optimization and Applications, 2008, pp. 286–295.
[4] Zhipeng Cai, Randy Goebel, Guohui Lin, Size-constrained tree partitioning: approximating the multicast k-tree routing problem, Theoret. Comput. Sci. 412 (3) (2011) 240–245.
[5] Zhipeng Cai, Guohui Lin, Guoliang Xue, Improved approximation algorithms for the capacitated multicast routing problem, in: Proceedings of 11th Annual International Conference on Computing and Combinatorics 2005, 2005, pp. 136–145.
[6] Moteiv Corporation, Telos (rev b): preliminary datasheet, http://www2.ece.ohio-state.edu/~bibyk/ee582/telosMote.pdf, 2004.
[7] Roland de Haan, Richard J. Boucherie, Jan-Kees van Ommeren, The impact of interference on optimal multi-path routing in ad hoc networks, in: Managing Traffic Performance in Converged Networks, in: Lecture Notes in Computer Science, vol. 4516, Springer, Berlin, Heidelberg, 2007, pp. 803–815.
[8] Budhaditya Deb, Sudeept Bhatnagar, Badri Nath, Reinform: reliable information forwarding using multiple paths in sensor networks, in: Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks, 2003.
[9] Yong Ding, Yang Yang, Li Xiao, Multi-path routing and rate allocation for multi-source video on-demand streaming in wireless mesh networks, in: Proceedings of IEEE INFOCOM, 2011, pp. 2051–2059.
[10] Deepak Ganesan, Ramesh Govindan, Scott Shenker, Deborah Estrin, Highly-resilient energy-efficient multipath routing in wireless sensor networks, in: Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing, MobiHoc, 2001.
[11] Faiza Iqbal, Muhammad Younus Javed, Anjum Naveed, Interference aware multi-path routing in wireless networks, in: 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS, Sept. 2008, pp. 516–518.
[12] Kamal Jain, Jitendra Padhye, Venkata N. Padmanabhan, Lili Qiu, Impact of interference on multi-hop wireless network performance, in: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, MobiCom, 2003, pp. 66–80.
[13] Yusuke Kobayashi, Induced disjoint paths problem in a planar digraph, Discrete Appl. Math. 157 (15) (2009) 3231–3238.
[14] Marc R. Pearlman, Zygmunt J. Haas, Peter Sholander, Siamak S. Tabrizi, On the impact of alternate path routing for load balancing in mobile ad hoc networks, in: First Annual Workshop on Mobile and Ad Hoc Networking and Computing, MobiHOC, 2000, pp. 3–10.
[15] Marjan Radi, Behnam Dezfouli, Kamalrulnizam Abu Bakar, Shukor Abd Razak, Hwee-Pink Tan, IM2PR: interference-minimized multipath routing protocol for wireless sensor networks, Wirel. Netw. 20 (7) (2014) 1807–1823.
[16] Jenn-Yue Teo, Yajun Ha, Chen-Khong Tham, Interference-minimized multipath routing with congestion control in wireless sensor network for high-rate streaming, IEEE Trans. Mob. Comput. 7 (9) (Sept. 2008) 1124–1137.

Please cite this article in press as: K. Zhang et al., DOAMI: A distributed on-line algorithm to minimize interference for routing in wireless sensor networks, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.05.043

[17] Thiemo Voigt, Adam Dunkels, Torsten Braun, On-demand construction of non-interfering multiple paths in wireless sensor networks, in: Proceedings of the 2nd Workshop on Sensor Networks at Informatik, 2005.
[18] Zijian Wang, E. Bulut, B.K. Szymanski, Energy efficient collision aware multipath routing for wireless sensor networks, in: IEEE International Conference on Communications, ICC, June 2009, pp. 1–5.
[19] Kui Wu, Janelle Harms, Performance study of a multipath routing method for wireless mobile ad hoc networks, in: Proceedings of the 9th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001, pp. 99–107.
[20] Guoqiang Yan, Weijun Duan, Chao Ma, Liang Huang, Minimize interference while using multipath transportation in wireless multimedia sensor networks, in: Recent Advances in Computer Science and Information Engineering, in: Lecture Notes in Electrical Engineering, vol. 127, Springer, Berlin, Heidelberg, 2012, pp. 239–244.
[21] Jie Zhang, Choong Kyo Jeong, Goo Yeon Lee, Hwa Jong Kim, Cluster-based multi-path routing algorithm for multi-hop wireless network, Int. J. Future Gener. Commun. Netw. 1 (2007) 67–75.
[22] Kejia Zhang, Qilong Han, Zhipeng Cai, Guisheng Yin, Junyu Lin, Metric and distributed on-line algorithm for minimizing routing interference in wireless sensor networks, in: The 9th Annual International Conference on Combinatorial Optimization and Applications, 2015, pp. 279–292.
[23] Kejia Zhang, Qilong Han, Guisheng Yin, Haiwei Pan, OFDP: a distributed algorithm for finding disjoint paths with minimum total length in wireless sensor networks, J. Comb. Optim. (2015) 1–19.

Please cite this article in press as: K. Zhang et al., DOAMI: A distributed on-line algorithm to minimize interference for routing in wireless sensor networks, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.05.043