Localized distance-sensitive service discovery in wireless sensor networks

(Technical Report)

Xu Li†, Nicola Santoro† and Ivan Stojmenovic‡

†SCS, Carleton University, Canada
Email: {xlii, santoro}@scs.carleton.ca

‡SITE, University of Ottawa, Canada
Email: ivan@site.uottawa.ca

***Abstract*—We propose a localized distance-sensitive service discovery protocol, *iMesh*, for wireless sensor networks. iMesh is a lightweight algorithm that uses no global computation and generates constant per node storage load. In iMesh, new service providers publish their location information in four directions, like in a mesh. The information propagation for relatively remote service is restricted by a blocking rule, which also updates the mesh structure. Based on an expansion rule, nodes along mesh edges may further advertise newly arrived relatively near service by backward distance-limited transmissions, replacing previously closer service location. Transmission paths form a planar structure, *information mesh*, which serves as service directory. Service consumers conduct a lookup process restricted within their home mesh cells to discover nearby services. We first present iMesh and study its properties over a grid network model. We then evaluate its performance in randomized sensor network scenarios through extensive simulation. Simulation results show that iMesh guarantees nearby service selection with very high probability ($> 99\%$) and with considerably low message overhead.**

I. Introduction

A. Framework

A wireless sensor network (WSN) is a collection of micro-sized wireless sensing devices, *sensors*, deployed in a region of interest for object monitoring and/or target tracking. In traditional WSNs, nodes are responsible only for sampling their surroundings and reporting to a pre-defined data sink. As hardware technology advances, WSNs are now evolving towards service-oriented networks such as mobile sensor networks [15], [23] and wireless sensor and actor networks [2], [16], where some service providers, i.e., mobile sensors or actors, offer movement-assisted services to other nodes and/or to the physical world. In mobile sensor networks, redundant sensors can geographically relocate to replace failed ones; in wireless sensor and actor networks, actors may move to a target location to, for example, deploy, repair or collect sensors, turn off water tap in cast of water-overflow, extinguish a fire in a woody area, interact with patients in a health care system, rescue survivors in an emergency situation, and so on.

Service discovery is a crucial component of any service-oriented network. Discovery criteria depend on the underlying network and the application. In the movement-assisted service delivery cases mentioned above, delivery distance is a primary concern for the purpose of energy-saving and timely response. Furthermore, since we are in the context of WSNs, service discovery must be performed in an efficient way, i.e., with constant storage load on each node and with no global computation. Although many specialized service discovery algorithms [5], [7], [9]–[12] and adoptable techniques [1], [4], [13], [14], [18], [20], [21] have been proposed for wireless ad hoc networks, they have different weaknesses. In particular, they are not suitable for the problem of *distance-sensitive* service discovery in resource-constrained WSNs, where the algorithms are expected to provide closest/nearby service selection guarantee:

***Definition** 1 (Closest Service Selection):* A service consumer discovers the closest service provider.

***Definition** 2 (Nearby Service Selection):* A service consumer discovers a service provider that is at most twice as far as the closest one.

Intuitively, if we construct a *Voronoi diagram* using service providers as creating points and let each of them distribute its location information along the perimeter of its Voronoi polygon, then the Voronoi diagram becomes a distributed service directory with bounded per-node storage load. In this case, distance-sensitive service lookup becomes localized. That is, a service consumer queries along a path in an arbitrary direction, and it will find its closest service provider once it hits the perimeter of its home Voronoi polygon. This intuitive solution possesses all the properties that we are looking for, but it requires *global computation*. Hence, to make this solution practical, as service directory we must substitute the Voronoi diagram with a localized planar structure that has good proximity. A naive idea of improvement is to replace Voronoi digram with *square mesh*. That is, service providers propagate their location information horizontally and vertically; the propagation paths form a mesh structure as service directory. Although this method requires only local computations, it can generate inconstant storage load on nodes if service providers are all placed in a line, and it has no guarantee on closest/nearby service selection because the mesh structure bears *no proximity* property.

As we show in this paper, it is however possible to modify the mesh-construction technique to obtain a planar structure that (as the mesh but unlike the Voronoi diagrams) can be constructed in a purely localized manner and (as the Voronoi diagram but unlike the mesh) possesses our required guaranteed proximity property. The needed modification is the use of distance-based blocking, and has been recently and independently proposed ([15], [22]), with little [15] or no [22] analysis of the resulting structure and its properties.

B. Contributions

In this paper, we propose a new localized distance-sensitive service discovery protocol, *iMesh*, for wireless sensor networks. The proposed protocol is based on the planar structure (called *information mesh* in [15]) created by the use of blocking in the traditional mesh-construction, as well as the use of a new information expansion rule.

In iMesh, service providers publish their location information, like in a mesh, in four directions: north, west, south and east. During their transmission, these information collinearly or orthogonally block each other, according to the blocking rule: a node receiving information from multiple service providers forwards only the information of the closest one. They may however also be extended to other directions according to the expansion rule: a node where information x orthogonally blocks information y transmits x along the backward transmission path of y for a limited distance. The transmission paths together constitute the information structure that distributedly stores the location information of all the service providers. To discover nearby services, service consumers can simply conduct a cross lookup process within their residing mesh cells.

The properties of iMesh (construction cost and service guarantee) are analyzed over a grid sensor network model, for ease of understanding. The study focuses first on the theoretical analysis of the information structure constructed by the blocking rule alone (denoted by iMesh-A) and by the entire protocol (denoted by iMesh-B). The analysis of the performance of the protocol is then provided by extensive simulation of iMesh-A, iMesh-B and the well-known Quorum-based location service (Quorum for short) [21]. We comparatively evaluate their performance in randomly placed sensor networks. Our experimental results show that iMesh generates significantly lower message overhead than Quorum, and that iMesh-B guarantees closest service selection with probability $> 97\%$ and nearby service selection with probability $> 99\%$, noticeably improving the distance sensitivity of iMesh-A with negligible extra communication.

In summary, the main contributions of this paper include:

- identify the distance-sensitive service discovery problem in wireless sensor networks and propose a novel localized solution protocol iMesh;
- derive the message complexity and the constant per node storage load property of iMesh with formal proof of correctness for grid networks;

- generalize the cases that iMesh violates closest service selection and derive the upper bound of the distance sensitivity of iMesh in those cases;
- evaluate the performance of iMesh through extensive simulation on random networks and conclude that iMesh nearly provides nearby service selection guarantee.

The rest of the paper is organized as follows: Sec. II reviews some related work; Sec. III defines the network model; Sec. IV and V describe iMesh-A and analyze its properties; Sec. VI presents iMesh-B; Sec. VII provides comparative simulation study of iMesh-A, iMesh-B and Quorum with randomly placed sensors; Sec. VIII extends iMesh to multi-service scenarios; Sec. IX concludes the paper.

II. Related Work

Many service discovery algorithms have been proposed for wireless ad hoc networks. These algorithms can be categorized either as *directory-based approach* or as *directory-less approach*. The former, e.g. [10]–[12], use a well structured service directory to store service provider information and to facilitate service lookup. They usually require global computation such as clustering and dominating set formation for service directory construction and maintenance. The latter e.g. [5], [7], [9], do not maintain any special component but rely on periodical service advertisement and multicasting-/anycasting- based service lookup. Their execution often involves (limited) flooding operations. Because these existing algorithms can generate large message overhead and/or require inconstant storage space on sensor nodes, they are not suitable for resource-constrained wireless sensor networks. A detailed survey of existing service discovery algorithms can be found in [8], [17].

In addition to specialized service discovery algorithms, there exist other techniques, for example, data-centric storage schemes [4], [14], [18], [20] and location services [1], [13], [21], that can be adopted to solve the service discovery problem in wireless sensor networks. In the following, we will briefly review some of these related works.

Fang, Gao and Guibas proposed the landmark-based data storage and retrieval scheme [4]. This scheme constructs a Voronoi diagram over the network using some predefined landmark nodes as creating points and distributes its dual, the Delaunay triangulation, to all the nodes in the network. Based on the Delaunay triangulation, every nodes establish a shortest path tree rooted at its home landmark node. A data producer hashes the data (according to data type) to a certain Voronoi cell. Then it distributes the data along the shortest path in the shortest path tree rooted at the landmark node in that cell. A data consumer queries along the shortest path to the hash cell in the same shortest path tree. And it gets the data when it hits the storage path or reaches the hash cell. This scheme involves many global operations and provide no energy-efficiency.

Ratnasamy, Karp, Yin and Yu proposed the Geographic Hash Table (GHT) data-centric storage scheme [18]. In GHT, a node hashes an event to a unique location by event type and routes it to that location by a greedy-face combined routing protocol. A node, called home node, closest to the hash location will receive the event. A node periodically distributes its recorded events along the home perimeter enclosing the corresponding hash location in the planar graph used. A data consumer can get a type of event data either from the event's home node or home perimeter. The drawbacks of this scheme are that a node near the data source may have to travel a long distance to retrieve the data, and that bottleneck spots can occur when some types of data are frequently requested.

Sarkar, Zhu and Gao proposed the double-ruling information brokerage scheme [20]. Nodes are projected onto a virtual sphere, and each of them executes the scheme as if it was on the sphere. A data producer hashes the data (according to data type) to a unique point on the sphere and then distributes the data along a diameter circle, uniquely determined by the hash point and its own location, in both directions. Data of the same type from different producers is hashed to and aggregated at the same point on the sphere. A data consumer computes the hash point of the data and queries along a carefully selected diameter circle. In this way, it can intersect all the replication circles of the data and get its requested information. This scheme may generate bottleneck spot problem, because all the same type of data are routed to the same hash node. It also often generates relatively long update and search routes when the service is available nearby.

Li, Jannotti, De Couto, Karger and Morris proposed the Grid Location Service (GLS) [13]. This algorithm partitions the sensor field evenly into grids and constructs a quad-tree structure over the grids. Then it uses a hash function, designed on basis of the quad tree, to match each node (by ID) to a unique subset of location servers geographically distributed in the network. Every node updates all its location servers with its current location information. As a result, a node can find the location of any other node by querying one of the location servers of that node. This protocol requires pre-knowledge of the sensor field for the grid partition. It may generate large message overhead since location updates and location queries travel along zigzag lines.

Stojmenovic, Liu and Jia presented the quorum-based location service [21]. In this algorithm, a node sends its current position to all the nodes located in a "column" in the network. When a node a wants to find the location of another node b, it queries along a "row" in the network. This row intersects the columns of all the other nodes, including that of b. As the query message travels along the row, it picks the latest location information about b. After the message reaches the ends of the row, it is forwarded to b, which then replies a directly with correct location. Alternatively, intersection nodes may reply immediately if the information is sufficiently fresh. The main weakness of this protocol is that location update and discovery has to cross the entire network. In addition, if all the nodes are collinear along a column, every node has to store every other's location, generating large storage load.

The idea of blocking the construction of the mesh, used here in the blocking rule, has been independently suggested by Tchakarov and Vaidya [22], and by Li, Santoro and Stojmenovic [15]; their proposal did not however contain any (correct) theoretical analysis of the resulting information structure and its properties. Unlike the protocol proposed here, neither of them considers the use, in addition to blocking, of the expansion rule that we show leads to major improvements in the performance.

III. Model and Definitions

In this paper, we first consider a wireless sensor network where nodes are placed exactly at the intersection points of a grid structure. For this model we provide theoretical analysis and verify these results by simulation. We then consider randomized sensor placement in the field and verify and confirm our findings in this widely used setting, showing practicality of our approach.

Nodes are classified as *service providers (SPs)* or *service consumers (SCs)*. In practice, SCs may be the nodes that require services themselves or the nodes that require services on behalf of their monitored physical objects. SPs are scattered in the network at random. All the nodes, whether SPs or SCs, have the same communication radius. We denote such a network by $G(V, E)$ (or simply by G), where V and E represent the set of nodes and the set of edges in G, respectively. We use $\nu(G)$ to represent the number of SPs in G. By definition, $\nu(G) \leq n$ where $n = |V|$. Given G, $\nu(G)$ can be written as ν without ambiguity.

We require that all the nodes be aware of their own location through a localization system such as the Global Positioning System (GPS). We believe this requirement is reasonable because of the goal of wireless sensor networks. We assume the standard restrictions, i.e., total reliability, FIFO communication channel, bidirectional links and finite communication delay, which are commonly used in distributed computing [19].

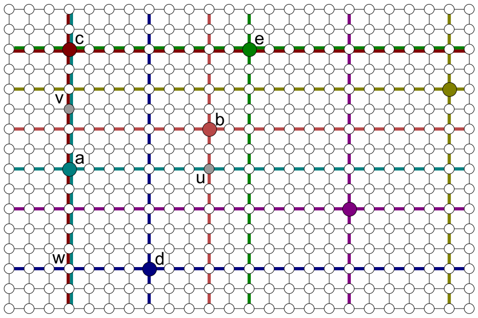
The reason for choosing to study first the grid network model is that we want to emphasis on the theoretical aspects of iMesh. In fact, iMesh can be implemented in arbitrary network scenarios by using GFG routing [3], [6] as part of the quorum-based location service [21]. We then use uniform random sensor networks in our simulation study to confirm theoretical findings.

IV. Basic iMesh Protocol

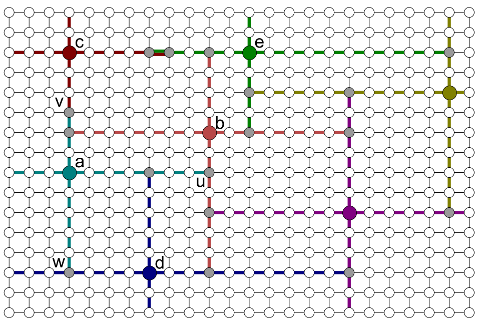
In this section, we will present the basic version (or version **A**) of iMesh, *iMesh-A*. We will first introduce how to build the service directory, i.e., information mesh, and then show how to conduct service lookup via it. The complete version of iMesh will be presented later, in Sec. VI.

A. Information mesh construction

Consider only the residing rows and columns of the SP-nodes in G. They intersect each other and constitute a complete mesh, as illustrated in



(a) A complete mesh



(b) An information mesh

Fig. 1. Information mesh construction

Fig. 1(a), where SP-nodes are represented by solid big dots, and their residing rows and columns are highlighted by thick lines. If each SP distributes its own location information (by a *registration message*) among the nodes along its residing row and column, this complete mesh distributedly stores the location information of all the SPs and therefore can be used for the purpose of service discovery.

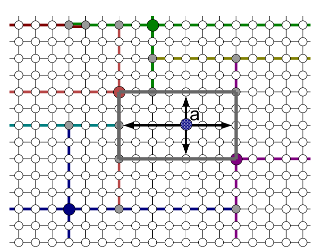
Let us closely examine the complete mesh structure in Fig. 1(a). SP-node c is closer to the area above the mid-point node v between itself and the vertically collinear SP-node a, and thus it has relatively high priority to be discovered by the SC-nodes in that area. In addition, SP-node b might be a better choice for the SC-nodes located in its right-side area than SP-node a. In these cases, a does not need to distribute its location information in those areas. Similar argument can be made against other SP-nodes. According to this observation, we define a blocking rule as follows:

***Rule** 1 (Blocking Rule):* For a node u shared by the residing rows/columns of two SP-nodes a and b ($a \neq b$), it stops the further propagation of a's registration message, if and only if $(|ua| > |ub|) \quad \vee \quad (|ua| = |ub| \wedge Collinear(a, b)) \vee (|ua| = |ub| \wedge \neg Collinear(a, b) \wedge Horizontal(b))$, where $Collinear(a, b)$ and $Horizontal(b)$ denote the case that a and b are (vertically or horizontally) collinear and the case that the involvement of b is along the horizontal direction, respectively. When this blocking happens, we say "b blocks a at u" and denote it by $a \overset{u}{\leftarrow} b$ or $b \overset{u}{\rightarrow} a$.

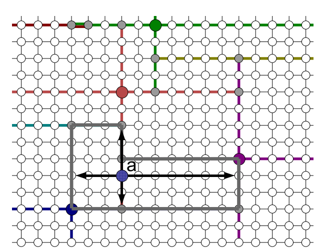
The application of above blocking rule can lead to the merger of adjacent mesh cells and result in a pruned mesh structure, which is the so-called *information mesh*. We denote the information mesh constructed on top of G by $\mathcal{M}(G)$ (or simply by $\mathcal{M}$). Figure 1(b), where solid small dots represent the nodes at which the blocking rule is applied, shows the information mesh corresponding to the complete mesh structure in Fig. 1(a). According to the definition of the blocking rule, we have the following corollary.

***Corollary** 1:* Information mesh is a planar structure.

In an asynchronous environment, a node c, at which a SP-node b is supposed to block another SP-node a, may wrongly retransmit the registration message of a, because of the late arrival of the registration message of b, violating the blocking rule. Fortunately, this problematic situation can be identified by c, as soon as it receives both of the two messages. Once c notices that, it as initiator starts a revocation process, in which the inconsistent information is erased from $\mathcal{M}$. More specifically, c sends a *revocation message* following the forward path of a's registration message. The revocation message is processed in exactly the same way as a registration message. It stops at a node where a's registration message stopped propagating. All the



(a) In-cell case



(b) On-edge case

Fig. 2. Cross lookup

nodes that receive this revocation message remove a's information from their local repositories. Such a revocation process can possibly lead to chain effect. That is, the registration message of a SP-node previously incorrectly blocked will now continue to propagate until the blocking rule is satisfied again.

B. Distance-sensitive service lookup

For a SC-node a, the objective of its service lookup is to identify the location of its target service provider $T(a)$ (see below for definition). According to the position of a, there are two possible lookup cases: *in-cell* case and *on-edge* case.

***Definition** 3 (Home Cell):* The home cell $HCell(a)$ of a SC-node a is the mesh cell where a is located in or the aggregation of the mesh cells which it is adjacent to.

***Definition** 4 (SPV):* The Set of Providers in Vicinity (SPV) of a SC-node a is the set of SPs that distribute their information along the perimeter of $HCell(a)$.

***Definition** 5 (Target Service Provider):* The target service provider $T(a)$ of a SC-node a is the nearest SP in a's SPV.

In the in-cell case, the SC-node a is located inside a cell of the information mesh. When a wants to find $T(a)$, it sends a search message along its residing grid row and column in four directions, as shown in Fig. 2(a), where the home cell of a is marked by thick gray lines and its search paths are highlighted by arrowed black lines. Such a search message stops its further transmission as soon as it hits a mesh edge (or the border of G), and then the node at which the message stops replies a with the location of its recorded SP-node closest to a (resp., a failure notice). If there is no SP in the network, what a will receive are all failure notice; otherwise, a can find the location of $T(a)$ simply by a local comparison among its received location data. Because the search paths of a form a cross, this service lookup method is called *cross lookup*.

The cross lookup method can also be applied to the on-edge situation, namely, when the SC-node a is riding on the information mesh. In this case, the search message that travels along a residing mesh edge of a will stop at the farthest end of the mesh edge on the home cell perimeter, as illustrated by Fig. 2(b). This way, a can reach all the composing mesh edges of its home cell and make a right decision.

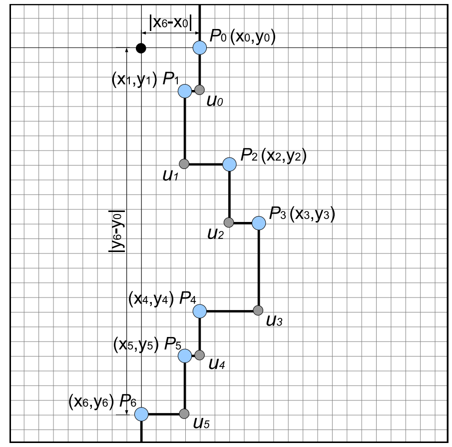


Fig. 3. Blocking chain $p_0 \overset{6}{\Leftarrow} p_6$ $(p_0 \overset{u_0}{\leftarrow} p_1 \overset{u_1}{\leftarrow} \cdots \overset{p_5}{\leftarrow} p_6)$

V. Analysis

In this section, we shall explore the theoretical aspects of iMesh-A and derive its properties. As we will see, iMesh-A has low message complexity and optimal per node storage load; however it does not always guarantee nearby service selection (rare counterexample cases exist).

***Definition** 6 (Chain Blocking):* For two SP-nodes a and b ($a \neq b$), b is said "chain-blocking a" if there is a blocking chain of length k ($k \geq 1$) from b to a, i.e., $a \overset{u_0}{\leftarrow} \cdots \overset{u_{k-1}}{\leftarrow} b$. We denoted this chain blocking by $a \overset{k}{\Leftarrow} b$ or $b \overset{k}{\Rightarrow} a$.

***Lemma** 1:* In a blocking chain $a \overset{k}{\Leftarrow} b$ along the Y axis (or X axis), the distance between a and b in X-direction (resp., Y-direction) is not longer than their distance in Y-direction (resp., X-direction).

Proof: Take as an example the blocking chain $p_0 \overset{6}{\Leftarrow} p_6$ in Fig. 3, where $a = p_0$, $b = p_6$ and $k = 6$. Consider two consecutive SP-nodes p_i and p_{i-1} ($1 \leq i \leq k$) in the chain. $|x_i - x_{i-1}| \leq |y_i - y_{i-1}|$, where (x_i, y_i) and (x_{i-1}, y_{i-1}) are respectively the coordinates of p_i and p_{i-1}. It is because that p_i, otherwise, can not block p_{i-1} in Y-direction. In this case, $|x_k - x_0| = |\sum_{i=1}^{k}(x_i - x_{i-1})| \leq \sum_{i=1}^{k}|x_i - x_{i-1}| \leq \sum_{i=1}^{k}|y_i - y_{i-1}| = |\sum_{i=1}^{k}(y_i - y_{i-1})| = |y_k - y_0|$. Thus the lemma holds. ■

***Definition** 7 (Extension):* The extension $\eta(\mathcal{M})$ (or η for brevity) of $\mathcal{M}$ is the length sum of the edges in $\mathcal{M}$.

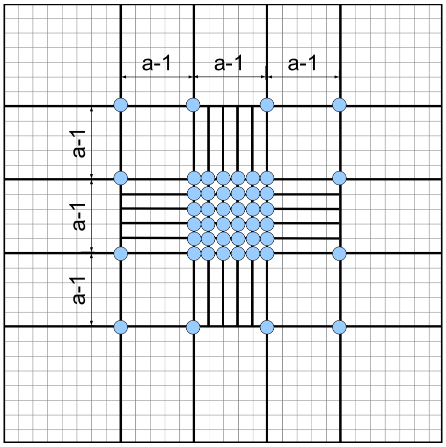


Fig. 4. An information mesh of $\eta = O(\nu + \sqrt{n})$

***Lemma** 2:* In a square G, $\eta \in O(Min\{\nu\sqrt{n}, n\})$.

Proof: For a complete mesh, as shown in Fig. 1(a), which is constructed without applying the block rule, its extension is just the multiplication of $\sqrt{n}$ and the number v of its constituting grid rows and columns of G. Clearly, the maximum value of v is 2ν, for example, in the case that there are no horizontally or vertically collinear SP-nodes. Therefore, the extension of the complete mesh is bounded above $O(\nu\sqrt{n})$. By the definition of the blocking rule, $\mathcal{M}$ is the result of edge pruning of the complete mesh structure, and therefore its extension is also bounded above $O(\nu\sqrt{n})$. This upper bounder is actually achievable, for example, when SP-nodes are all located on the same line along the X axis (or the Y axis). Note that, when $\nu > \sqrt{n}$, $\nu\sqrt{n}$ can be much larger than n in terms of order of magnitude. Furthermore, since $\mathcal{M}(G)$ is accommodated within G, its extension η obviously never exceeds $|V| = 2n - 2\sqrt{n} = O(n)$, the total number of edges in G. Hence, $\eta \in O(Min\{\nu\sqrt{n}, n\})$. ■

***Lemma** 3:* In a square G, $\eta = \Omega(\nu + \sqrt{n})$.

Proof: In $\mathcal{M}$, every SP-node has exactly four incidental edges, each of which is shared by at most two SP-nodes, and thus the number of edges is not less than 2ν. Under this circumstance, because each mesh edge has length at least 1, η is bounded below $O(\nu)$. Now, let us consider a northmost SP-node p_0. If p_0 is not blocked in Y-direction, its entire residing column will be included in $\mathcal{M}$; otherwise, there must exist a blocking chain spanning the entire network along the Y axis. In either case, η is not less than $\sqrt{n}$. Hence, by above analysis, $\eta = \Omega(\nu + \sqrt{n})$. This proves the lemma. ■

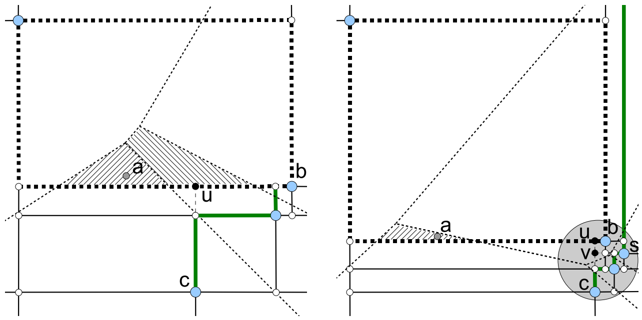
Note that, the lowerbound indicated by Lemma 3 is achievable, for example, in the scenario shown in Fig. 4. In this example, there are $\nu = a^2 + 12$ SP-nodes: a^2 are densely packed in the middle of G, constituting an $a \times a$ inner grid; 12 are evenly placed around the inner grid at distance $a - 1$, forming a big square that blocks the inner grid expanding. The length summation of the mesh edges is less than 6ν (in fact, it should be $6(\nu - a - 12)$) inside the big square; on the outside, it is less than $8\sqrt{n}$ (in fact, it should be $8(\sqrt{n} - 3a + 2)$). Thus in total is $\eta < 8\sqrt{n} + 6\nu = O(\nu + \sqrt{n})$.

***Theorem** 1:* In a square G, the message complexity of information mesh construction is $O(\psi(G))$, where $\nu + \sqrt{n} \leq \psi(G) \leq Min\{\nu\sqrt{n}, n\}$.

Proof: If G is a synchronous environment, the paths that SP-nodes' registration messages travel are exactly the edges of $\mathcal{M}$. In this case, due to the blocking rule, a constant number (1 or 2) of registration messages are transmitted on each communication link in these mesh edges. Specifically, there are two registration messages transmitted on the middle link of two collinear SP-nodes separated by an odd number of hops (as the case with c and e in Fig. 1(b)), and one registration messages over all the other links. Hence, the theorem follows immediately from Lemma 2 and 3. If, otherwise, G is an asynchronous environment, because some registration messages may be incorrectly transmitted on the links in $G - \mathcal{M}$, and revocation messages are used for consistency maintenance, the message complexity can not be lower than that in an synchronous scenario. On the other hand, because SP-nodes still block messages effectively, we can easily find there are at most 4 messages, 2 in each direction, transmitted on each link in the complete mesh structure, and thus the message complexity can not be worse than $O(Min\{\nu\sqrt{n}, n\})$. Hence, the theorem holds. ■

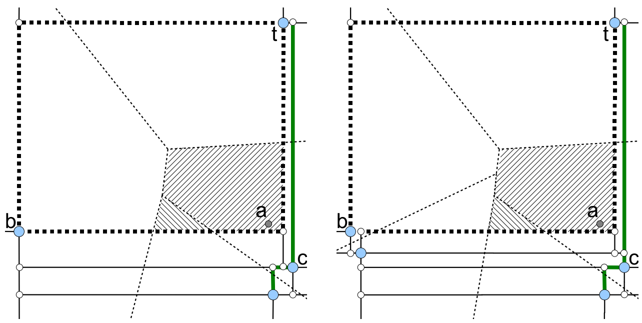
***Theorem** 2:* In a square G, the message complexity of cross lookup is $O(\sqrt{n})$.

Proof: A cross lookup process of a SC is restricted within a search cell, i.e., the home cell of the SC, which can be a single mesh cell or the aggregation of several mesh cells. In worst



(a) Barrage Case (b) Clean-Pass Case

Fig. 5. The example situations of $1 < TCR(a) \leq 2$



(a) Dirty-Pass Case (b) Isolation Case

Fig. 6. The example situations of $TCR(a) > 2$

case, for example, when SPs are all located on the same network border, a search cell spans the entire network, and a SC in the search cell will inquire all the way along its residing grid row and/or column, generating $O(\sqrt{n})$ search messages. This proves the theorem. ■

***Theorem** 3:* iMesh generates constant $O(1)$ storage load on each node.

Proof: Each of the nodes that constitute the information mesh records at most one SP-node's information from each of the four directions, i.e., the north, the south, the west and the east, due to the application of the blocking rule. As for the nodes which are not part of the information mesh, they do not store any data at all. Thus the theorem holds. ■

***Definition** 8 (Target over Closest Ratio):* The target over closest ratio $TCR(a)$ of a SC-node a is defined as $TCR(a) = \frac{|aT(a)|}{|aC(a)|}$, where $C(a)$ is a globally closest SP-node to a.

TCR measures the distance sensitivity of iMesh. Ideally, $TCR(a)$ is equal to 1, meaning closest service selection. This happens when the residing grid row and/or column of $C(a)$ is part of the perimeter of $HCell(a)$. However, due to the randomized distribution of SPs, it may not always be the case. To study the TCR performance (i.e., distance sensitivity) of iMesh-A, all the possible violation situations where $TCR > 1$ need to be identified. From a deep investigation, it is observed that all violations are virtually the variants of the following four basic cases:

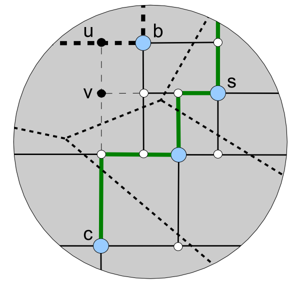


Fig. 7. An amplified version of the circled part in Fig. 5(b)

1) *Barrage case*: $C(a)$ is chain-blocked by a SP-node in a's SPV, before its blocking chain passes around $HCell(a)$;
2) *Clean-Pass case*: the blocking chain of $C(a)$ passes around $HCell(a)$ at the corner where a SP-node is located;
3) *Dirty-Pass case*: the blocking chain of $C(a)$ passes around $HCell(a)$ at the corner where no SP-node exists, and a composing mesh edge of $HCell(a)$ intersects the residing mesh edge of $C(a)$;
4) *Isolation case*: the blocking chain of $C(a)$ passes around $HCell(a)$ at the corner where no SP-node exists, and no composing mesh edge of $HCell(a)$ intersects the residing mesh edge of $C(a)$.

Figures 5 and 6, where irrelevant SC-nodes are hidden and SP-nodes are represented by solid big dots, illustrate above four basic violation cases. In the two figures, the home cell $HCell(a)$ of SC-node a is emphasized by broken thick lines, and the blocking chain of $c = C(a)$ is highlighted by complete thick lines; broken thin lines indicate the Voronoi diagram created using SP-nodes, and shadowed areas are the places where TCR ratio is greater than 1.

***Lemma** 4:* In Barrage case, $TCR(a) \leq 2$.

Proof: Suppose that b is the SP-node in a's SPV that chain-blocks c (i.e., $C(a)$). Then $|aT(a)| \leq |ab|$. Without loss of generality, assume that the chain of blocking happens along the Y axis, as shown in Fig. 5(a). By Lemma 1, $|bu| \leq |cu|$. Observe that angle $\angle cua$ can not be acute in any case. Thus ca is the longest side in triangle Δcua. Namely, $|cu| < |ca|$ and $|ua| < |ca|$. Then $|ab| \leq |bu| + |ua| \leq |cu| + |ua| < |ca| + |ca| = 2|ca|$. Recall $|aT(a)| \leq |ab|$. Hence, $|aT(a)| \leq 2|ca|$, which proves the lemma. ■

***Lemma** 5:* In Clear-Pass case, $TCR(a) \leq 2$.

Proof: Without loss of generality, assume that the blocking chain of c (i.e., $C(a)$) is towards $HCell(a)$ along the Y axis, as shown in Fig. 5(b). Examine the amplified version in Fig. 7. By Lemma 1, $|sv| \leq |vc|$. Unambiguously, $|bu| \leq |sv| \leq |cv| \leq |cu|$. From this point, the lemma then follows from the same proof as Lemma 4. ■

***Lemma** 6:* In both Dirty-Pass case and Isolation case, $TCR(a)$ may be larger than 2 but must not be larger than $\frac{d(G)}{|aC(a)|}$, where $d(G)$ is spatial the diameter of G.

Proof: Let us examine the scenarios given in Fig. 6, where $t = T(a)$ and the blocking chain of c (i.e., $C(a)$) is along the Y axis. By observation, $|at|$ is already greater than $2|ac|$, namely, $TCR(a) > 2$, and there is no restriction on the distance from t to the residing grid row of c. If we move b (together with d in Fig. 6(b)) and t far apart from a while maintaining their blocking relation, then $|at|$ could be way larger than $2|ac|$. On the other hand, because no pair of nodes has their separation larger than $d(G)$, we have $|at| \leq d(G)$ and consequently $TCR(a) = \frac{|at|}{|ac|} \leq \frac{d(G)}{|ac|}$. ■

VI. Complete iMesh Protocol

A distance-sensitive service discovery algorithm is expected to be able to ensure nearby service selection, that is that $TCR(a) \leq 2$ for any SC-node a. By Lemma 6, iMesh-A may unfortunately violate this expectation in Dirty-pass case and Isolation case. In this section, we will present the complete version (or version **B**) of iMesh, *iMesh-B*, which achieves major improvement on distance sensitivity over, but has the same complexity as, iMesh-A.

Define the *territory* of an arbitrary SP-node c as the area in which c can be discovered by the local SCs through the cross lookup method (refer to Sec. IV-B for cross lookup). The larger the territory of c, the higher its probability of being discovered, and thus the better the distance sensitivity of iMesh. However, in iMesh-A, the size of a SP's territory is strictly restricted by the blocking rule for message saving purpose. Figure 8 redraws the Dirty-Pass situation given in Fig. 6(a). In this figure, the registration paths of SP-node c are marked by arrowed hollow lines, and its territory is represented by the light gray area, which is the aggregation of the mesh cells adjacent by the registration paths of c.

In order to improve the distance sensitivity of iMesh, territory expansion is a must. In iMesh-B, the information mesh is built not only according to the blocking rule but also using an expansion rule. The new expansion rule enables SPs to expand their territories in the case of orthogonal blocking. The formal definition of the expansion rule is given below:

***Rule** 2 (Expansion Rule):* For a node u at which a SP-node a orthogonally blocks another SP-node b, it sends the location information of a to b along the backward path from which it receives b's information. The information of a does not travel all the way to b but stops at the point where the path intersects the bisector between a and b.

In Fig. 8, the transmission paths of the expansion messages of SP-node c is highlighted by arrowed solid lines, and the dark gray area is the expansion part of the territory of c. By observation, c's territory expands into the home cell $HCell(a)$ of SC-node a, and a becomes able to discover c as a result. Consider another SC-node a' who shares the same home cell with a. The closest SP-node $C(a')$ of a' is d in the blocking chain of c. In iMesh-A without the application of the expansion rule, $TCR(a')$ could be way greater than 2 (if $HCell(a)$ is very large) according to Lemma 6. On the contrary, after the expansion rule is applied in iMesh-B, we have $T(a') = c$ and then $TCR(a') \leq 2$ following a similar proof as Lemma 4.

By above examples, the expansion rule effectively eliminates the Dirty-Pass case, and thus Lemma 6 only partially holds for iMesh-B. By definition, the expansion rule does not either change the structure of, or remove any location information from, the information mesh. Therefore, Lemma 2, 3, 4 and 5

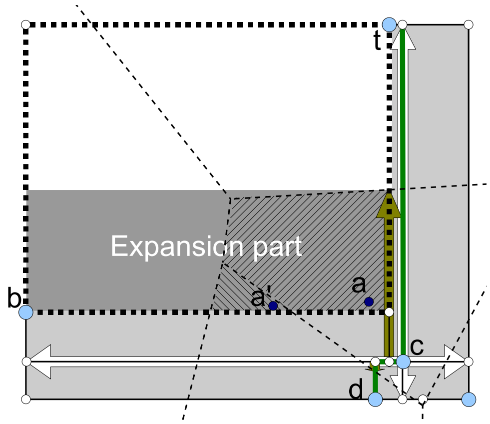


Fig. 8. The effect of the expansion rule

and Theorem 2 still hold for iMesh-B. In addition, it is not difficult to verify that Theorem 1 and 3 are also applicable to iMesh-B. In summary, the expansion rule actually enables iMesh-B to achieve improved overall distance sensitivity over iMesh-A at very low cost. Its effect and cost will be seen clearly later, through simulation in Sec. VII.

VII. PERFORMANCE EVALUATION

As summarized in Sec. II, existing service discovery algorithms and adoptable techniques usually rely on global computation and therefore generate large message overhead, and they may in addition impose inconstant storage load on network nodes and/or induce bottleneck problem in the network. Our proposed protocol iMesh however has obvious advantages in all these aspects. It aims to yield optimal (constant) per node storage load and avoid long service registration/lookup paths while providing satisfactory distance-sensitivity.

In the case that no comparable work actually exists, we evaluate iMesh in comparison with the well-known quorum-based location service (Quorum for short) [21], through an extensive set of simulation. As we will see in the following, iMesh has considerably low message overhead compared with Quorum, and iMesh-B guarantees closest service selection with high probability, larger than 97%, and nearby service selection with very high probability, larger than 99%, significantly impoving the distance sensitivity of iMesh-A at negligible communication cost.

A. Evaluation metrics

We study the message overhead of iMesh in comparison with Quorum's using the following metrics:

- *Total Number of Construction Messages (TNCM)*: the total number of messages transmitted in the network for information mesh construction;
- *Number of Construction Messages per SP (NCMSP)*: the average number of messages generated in the network by an arbitrary SP for the purpose of information mesh construction;
- *Number of Search Messages per SC (NSMSC)*: the average number of service lookup messages generated in the network by an arbitrary SC (reply messages are not counted);

As Quorum guarantees closest service selection, the following metrics of distance sensitivity evaluation are for iMesh only:

- *Average TCR* and *Peak TCR*: the average TCR and the maximum TCR of all the possible SCs in the network.
- Probabilities of $TCR = 1$, $1 < TCR \leq 2$, and $TCR > 2$ (*PTCR1*, *PTCR2*, and *PTCR3*): the probability that the TCR of an arbitrary SC in the network satisfies the corresponding condition.

B. Simulation setup

We simulated iMesh-A, iMesh-B and the Quorum within a custom java-based network simulator, which uses the Greedy-Face-Greedy (GFG) routing technique [3], [6] to support directional message transmission (for the detail on how, one can refer to [15], [21]). Our simulation was carried out over a large-scaled sensor network that contains $10,000$ nodes and fully covers the sensor field. The average node density is $8-9$. The scenarios with larger node densities are also tested, and the results are even better (because less nodes are involved in message transmissions) and therefore will not be presented.

We run two sets of experiments. In the first set, the network is set to be a synchronous environment with simultaneous execution and unified link delay; in the second set, the network is configured to be an asynchronous environment where SP-nodes start the protocols maximally 30 simulated time units off each other, and each communication link has transmission delay of 10 simulated time units at most. We choose the settings with the percentage of SPs (PSP) in the network varying from 1% to 50%. For each setting, we executed iMesh-A, iMesh-B and Quorum over 100 randomly generated network scenarios to get average results.

As a matter of fact, we as well conducted experiments in the settings where PSP is larger than 0 and less than 1%. We find that Quorum outperforms iMesh in service registration overhead (TNCM/NCMSP) with very small PSP ($< 0.3\%$ in iMesh-A; $< 0.2\%$ in iMesh-B). The reason is quite obvious: the cross registration paths used in iMesh lead to more messages in total when message blocking happens at a very low frequency. These easily-understood results are displayed in the Appendix without further explanation.

C. Experimental results

In this subsection, we are going to present our average experimental results, whose confidence interval is within 10%. We will first closely examine the message overhead of iMesh and then investigate its distance sensitivity in detail.

1) Message overhead: We first study the performance difference between the two versions of iMesh in a synchronous environment and in an asynchronous environment.

Figure 9(a) show the TNCM of iMesh in relation with PSP. For reference, mesh *extension* (Definition 7) is also drawn in the figure. As PSP grows, the information mesh has a more and more complex structure and is therefore expected to exhibit an increasing extension and a growing construction message overhead. The expectation is confirmed by the ascending trend of the curves in the figure. The small gap between the TNCM curves for iMesh-A and iMesh-B in either environment indicates that the overhead of the *expansion rule* (Rule 2) is minor. And, from the figure we can also see that TNCM will never exceed some constant times mesh extension. This observation verifies Theorem 1.

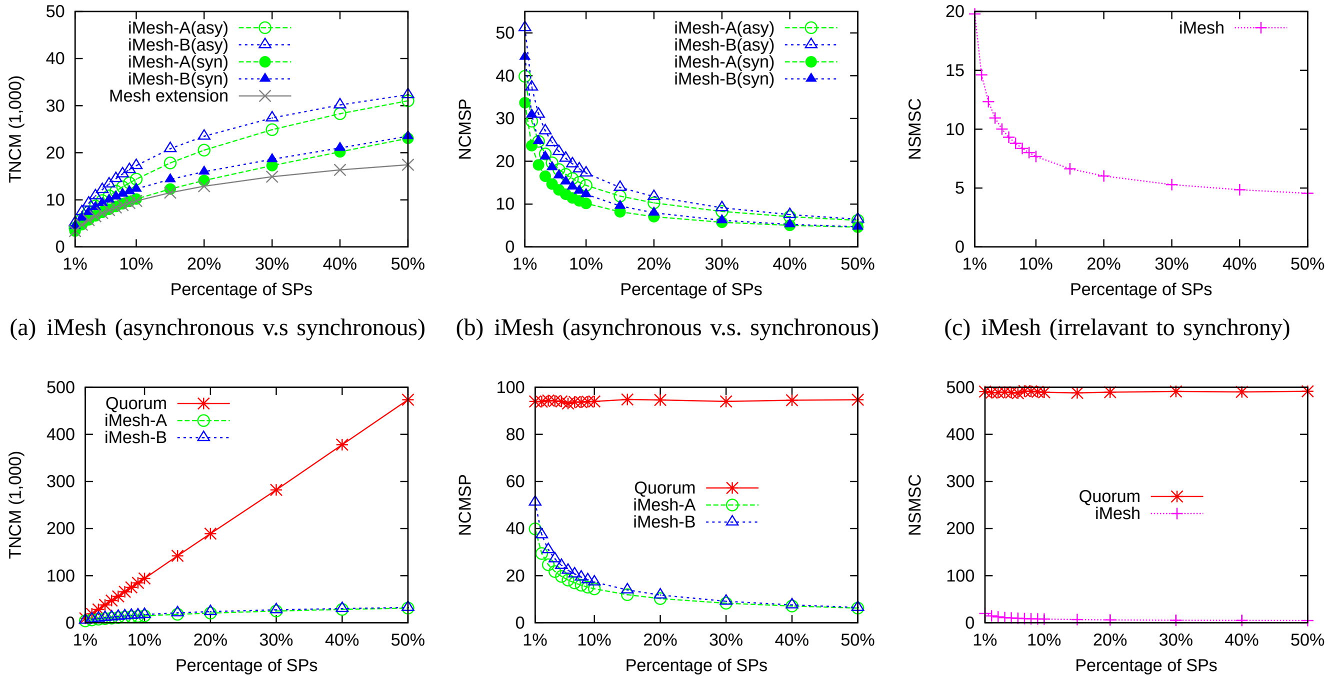
Examine again Fig. 9(a) and pay attention to the difference of TNCM in the two environments. It is observed that TNCM is always higher in the asynchronous environment than in the synchronous environment. This is due to the extra messages used for eliminating the information inconsistency caused by asynchrony. Further, as PSP grows in either environment, TNCM curves deviate more and more from the curve of mesh extension, and the TNCM of iMesh-B approaches to the that of iMesh-A closer and closer. It is because, when there are more SP-nodes, the situation that two collinear SP-nodes are an odd number of hops away happens more often, causing more overlapping registration messages on mesh edges, and the mesh cell has smaller size, leading to the reduction of the travel distance of expansion messages.

Figure 9(b) displays the NCMSP of iMesh as a function of PSP. From the figure, we can see that NCMSP drops and approaches to 4 as PSP goes up. It is because, when SP density increases, a SP-node's registration message travels a decreased hop-distance (on average) in each direction before being blocked, and the travel distance can be as low as 1-hop, resulting in merely 4 registration messages in the extreme case. As shown in the figure, each SP-node uses slightly more construction messages in the asynchronous environment than in the synchronous environment due to the cost of information consistency maintenance; iMesh-B generates slightly larger NCMSP than iMesh-A in both environments, which again implies the negligible message cost of the expansion rule.

Figure 9(c) depicts the NSMSC of iMesh, which is irrelevant to synchrony and to the application of the expansion rule, as a result of PSP. It is observed that NSMSC drops and approaches to 4 as PSP climbs. It is because, when SP density increases, a SC-node's search message travels a decreased hop-distance (on average) in each direction before finding a SP, and the travel distance can be as low as 1-hop, resulting in merely 4 search messages in the extreme case.

As Quorum is irrelevant to network synchrony, we will study the performance difference of iMesh and Quorum in an asynchronous environment.

Figure 9(d) shows Quorum v.s. iMesh in TNCM in relation with PSP. We can observe that, as PSP increases, TNCM goes up quickly in Quorum, while it climbs at a very slow speed in iMesh (nearly 5 times slower than in Quorum), starting almost from the same point. It is because, a SP's registration message always propagate across the entire network in Quorum; but, as confirmed by Fig. 9(b), it how-



(a) iMesh (asynchronous v.s synchronous) (b) iMesh (asynchronous v.s. synchronous) (c) iMesh (irrelavant to synchrony)

(d) iMesh (asynchronous) v.s. Quorum (e) iMesh (asynchronous) v.s. Quorum (f) iMesh v.s. Quorum

Fig. 9. Message overhead

ever travels a shorter and shorter distance due to message blocking in both version of iMesh when PSP goes up.

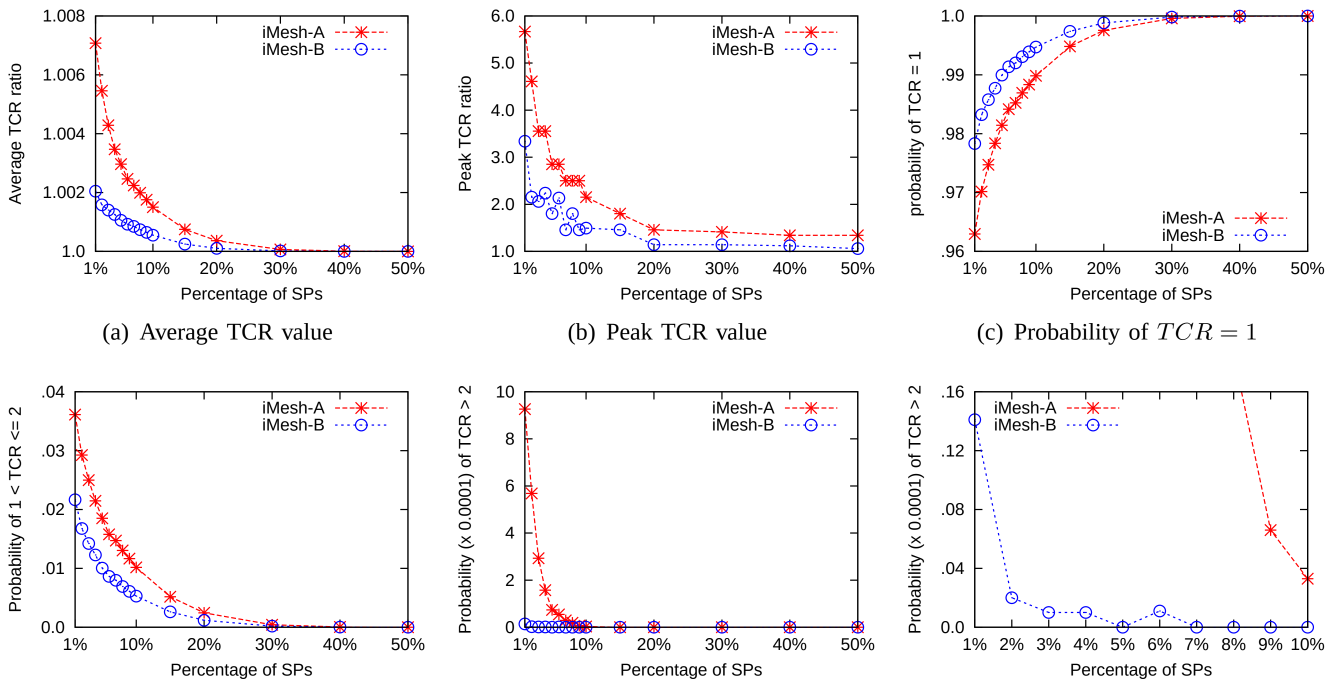
Figure 9(e) shows the NCMSP of Quorum and of iMesh as a result of PSP. We can observe that the curve corresponding to Quorum is nearly a horizontal line. It is because in Quorum a SP's registration message has to travel across the entire network, whose width is constant. On the contrary, iMesh (both versions) has very low NCMSP due to message blocking, and as we explained in previous paragraph, the larger PSP, the more often message blocking happens, and therefore the lower the NCMSP. This figures shows in a detailed level that Quorum is nearly 5 times expensive when $PSP \geq 10\%$, and at least 2 times expensive when $1\% \leq PSP < 10\%$, than iMesh.

Figure 9(f) shows iMesh versus Quorum in NSMSC as PSP varies. From this figure we can observe that Quorum generates almost constant NSMSC regardless of PSP, and that the NSMSC of Quorum is dramatically larger (over $20-100$ times larger) than that of iMesh. It is because, a SC in Quorum has to search across the entire network and along the whole outer boundary for a closest SP; while in iMesh, a SC does not query along the outer boundary of the network, and its service lookup operation (i.e., cross lookup) is restricted within a search cell, whose size decreases as PSP increases.

To sum up, the results given in Figure 9 clearly indicate that protocol iMesh (whether the A version or the B version) use a considerably small number of messages for service registration and service lookup, considering network size and compared with the Quorum algorithm [21]. In a detailed level, iMesh-B generates slightly larger message overhead than iMesh-A; but the difference is actually negligible.

2) Distance sensitivity: We shall now proceed to the distance sensitivity evaluation of iMesh. Before moving further, we would indicate that the results to be presented below are regardless of the (synchronous or asynchronous) nature of the execution environment.

Figures 10(a) and 10(b) respectively show the average TCR and the peak TCR in relation with PSP. From Fig. 10(a) we can see that the average TCR is nearly equal to 1 in all the PSP cases. This is because of the low probability of $TCR > 1$. In



(a) Average TCR value (b) Peak TCR value (c) Probability of $TCR = 1$

(d) Probability of $1 < TCR \leq 2$ (e) Probability of $TCR > 2$ (f) Large view of 10(e) for PSP in 1% - 10%

Fig. 10. Distance sensitivity

both of the two figures, the curves decline and approach to 1 closer and closer as PSP increases. This phenomenon is due to the decreasing probability of $TCR > 1$. According to the two figures, iMesh-B always has better distance sensitivity than iMesh-A. It is because the expansion rule (Rule 2) effectively eliminates the Dirty-pass case.

Figures 10(c) – 10(e) depict PTCR1, PTCR2 and PTCR3 as a function of PSP, respectively. By Fig. 10(c), both iMesh-A and iMesh-B provides closest service selection with high probability, respectively larger than 96% and 97%. By Fig. 10(d) and 10(e), both PTCR2 and PTCR3 quickly drop down nearly to 0 as soon as the density of service providers increases to 10%. The three figures together indicate that iMesh guarantees nearby service selection with very high probability, larger than 99%, in all PSP cases, and they also confirm our analysis about TCR value in previous paragraph.

Figures 10(c) – 10(e) also imply that iMesh-B always has better distance sensitivity than iMesh-A. It is because iMesh-B eliminates the Dirty-Pass case by the expansion rule. Examine the part for PSP in range from $1\% - 10\%$ in Fig. 10(e). An amplified version of this part is given in Fig. 10(f). We can see that the PTRC3 of iMesh-A and iMesh-B are both extremely low, i.e., smaller than $O(10^{-4})$. In particular, due to the application of the expansion rule, iMesh-B's PTCR3 is significantly lower, in terms of oder of magnitude, than that of iMesh-A.

The experimental results displayed in Fig. 10 indicate that protocol iMesh (whether version **A** or version **B**) has satisfactory distance sensitivity. More specifically, iMesh-B performs much better in both closest service selection aspect and nearby service selection aspect and has lower probability of undesired distance service selection, when compared with iMesh-A. According to Sec. VII-C1, we can find that iMesh-B in fact achieves these advantages over iMesh-A at the negligible cost of messages.

VIII. Extension to Multi-service Scenarios

In previous sections, iMesh was presented under the assumption of single-service networks, which is however not a common setting in practice. When there is more than one type of service provided in the network, a multi-layered information mesh can be constructed to support service discovery. That is, the same type of service providers together constitute a mesh layer, and different layers correspond to different types of services. For a network with $k \geq 1$ service types, the height, i.e., the number of layers, of the information mesh is equal to k. Note that a node offering multiple types of services will appear in more than one layer of the information mesh. Fig. 11 shows an information mesh of three layers. In this figure, SP-node p offers all the three type of services and thus exists in every single layer of the information mesh.

With cares, the message complexity of constructing a multi-layered information mesh can be made not larger than the summation of the message complexity of building every single mesh layer separately. For instance, a SP-node shared by t number of mesh layers does not necessarily distribute its location information t times. Instead, it attaches t bits to the location information to indicate its offered services. The information is virtually blocked in one layer by flipping the corresponding bit; it physically stops propagating when all the attached bits are flipped or when it reaches the network border. By this means, the SP-node fulfills its construction duty in all its residing layers simultaneously, thus saving a considerable number of messages. An obvious coarse upper bound of the message complexity is $\nu\sqrt{n}$. The study of the precise message complexity is however not included in our current work.

With the multi-layered information mesh, when a node wants to discover a particular type of service, it just needs to perform cross service lookup in the corresponding layer as if it is still in a single-service network. In this way, the distance sensitivity and the service lookup message overhead of iMesh naturally stay unchanged. Because a node shared by t $(1 \leq t \leq k)$ mesh layers has to store a constant amount of information for each of its residing layer (by Theorem 3), it has $O(t) \leq O(k)$ storage load in total. Since k is usually a known value at the network deployment time, iMesh still yields constant per node storage load.

IX. Conclusions

Swift developing hardware technology favors the evolvement of wireless sensor networks (WSNs) toward service-oriented networks and renders WSN-based service discovery a rising research issue.

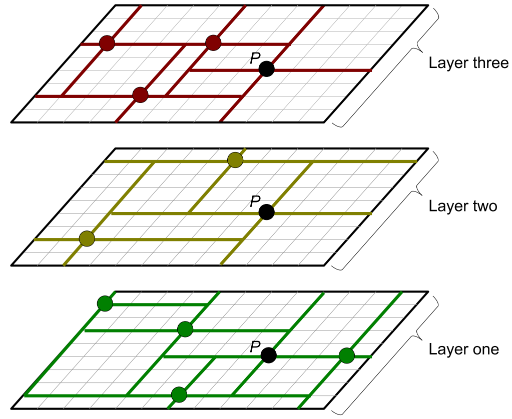


Fig. 11. A 3-tier information mesh

Some emerging service-oriented wireless sensor networks, for example, mobile sensor networks and wireless sensor and actor networks, indicate the particular need in distance-sensitive service discovery algorithms. Although many specialized service discovery approaches and adoptable techniques have been proposed for wireless ad hoc networks in the literature, they have weaknesses in message overhead and storage load and are thereby not a good option for wireless sensor networks with severe resource constraints.

In this paper, we proposed a novel distance-sensitive service discovery protocol, iMesh, for WSNs. iMesh is grounded on a localized planar structure, i.e., information mesh, which possesses good proximity property. We comprehensively and detailedly studied the properties of iMesh from theoretical point of view. Our analysis show that iMesh has very low message complexity and constant per node storage load. Through extensive simulation we studied the performance of iMesh in comparison with the Quorum-based location service [21]. Simulation results verify our theoretical findings and indicate that, iMesh nearly provide closest/nearby service selection guarantee with message overhead way smaller than Quorum's. Particularly, they indicate that iMesh-B outperforms iMesh-A in both closest service selection aspect and nearby service selection aspect practically at no cost.

Acknowledgements

This work was partially supported by NSERC grants.

REFERENCES

[1] I. Abraham, D. Dolev, and D. Malkhi. "LLS: a Locality Aware Location Service for Mobile Ad Hoc Networks". In *Proceedings of the Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, pages 75–84, 2004.

[2] I. F. Akyildiz and I. H. Kasimoglu. "Wireless sensor and actor networks: research challenges". *Ad Hoc Networks Journal (Elsevier)*, 2(4):351–367, 2004.

[3] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks". In *Proceedings of the 3rd ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM)*, pages 48–55, 1999.

[4] Q. Fang, J. Gao, and L. J. Guibas. "Landmark-Based Information Storage and Retrieval in Sensor Networks". In *Proceedings of the 25th Conference of the IEEE Communications Society (INFOCOM)*, pages 286–297, 2006.

[5] C. Frank and H. Karl. "Consistency Challenges of Service Discovery in Mobile Ad Hoc Networks". In *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pages 105–114, 2004.

[6] H. Frey and I. Stojmenovic. "On Delivery Guarantees of Face and Combined Greedy-Face Routing Algorithms in Ad Hoc and Sensor Networks". In *Proceedings of the 12th Annual ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 390–401, 2006.

[7] Z. Gao, L. Wang, M. Yang, and X. Yang. "CNPGSDP: An Efficient Group-based Service Discovery Protocol for MANETs". *Computer Networks (Elsevier)*, 50(16):3165–3182, 2006.

[8] Z. Gao, Y. Yang, J. Zhao, J. Cui, and X. Li. "Service Discovery Protocols for MANETs: A Survey". In *Proceedings of MSProceeding of the 2nd International Conference on Mobile Ad-hoc and Sensor Networks (MSN) (LNCS 4325)*, pages 232–243, 2006.

[9] S. Helal, N. Desai, V. Verma, and C. Lee. "Konark - A Service Discovery and Delivery Protocol for Ad-Hoc Networks". In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2107–2133, 2003.

[10] M. Klein and B. K. Ries. "Multi-layer Clusters in Ad Hoc Networks - an Approach to Service Discovery". In *Proceedings of Networking Workshops on Web Engineering and Peer-to-Peer Computing (LNCS 2376)*, pages 99–110, 2002.

[11] M. Klein, B. K. Ries, and P. Oberiter. "Service rings - a Semantic Overlay for Service Discovery in Ad Hoc Networks". In *Proceedings of th 14th IEEE International Workshop on Database and Expert Systems Applications (DEXA)*, pages 180–185, 2003.

[12] U. C. Kozat and L. Tassiulas. "Service Discovery in Mobile Ad Hoc Networks: an Overall Perspective on Architectural Choices and Network Layer Support Issues". *Ad Hoc Networks Journal (Elsevier)*, 2(1):23–44, 2004.

[13] J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, and R. Morris. "A Scalable Location Service for Geographic Ad Hoc Routing". In *Proceedings of the 6th Annual ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 120–130, 2000.

[14] X. Li, Y. J. Kim, R. Govindan, and W. Hong. "Multi-dimensional Range Queries in Sensor Networks". In *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 63–75, 2003.

[15] X. Li, N. Santoro, and I. Stojmenovic. "Mesh-based Sensor Relocation for Coverage Maintenance in Mobile Sensor Networks". In *Proceedings of the 4th International Conference on Ubiquitous Intelligence and Computing (UIC) (LNCS 4611)*, pages 696–708, 2007.

[16] T. Melodia, D. Pompili, V. C. Gungor, and I. F. Akyildiz. "A Distributed Coordination Framework for Wireless Sensor and Actor Networks". In *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 99–110, 2005.

[17] A. N. Mian, R. Beraldi, and R. Baldoni. "Survey of Service Discovery Protocols in Mobile Ad Hoc Networks". Technical Report 4/06, Universit degli Studi di Roma La Sapienza, Rome, Italy, 2006.

[18] S. Ratnasamy, B. Karp, L. Yin, and F. Yu. "GHT: A Geographic Hash Table for Data-Centric Storage". In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, pages 78–87, 2002.

[19] N. Santoro. *Design and Analysis of Distributed Algorithms*. Wiley, 2007.

[20] R. Sarkar, X. Zhu, and J. Gao. "Double Rulings for Information Brokerage in Sensor Networks". In *Proceedings of the 12th Annual ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 286–297, 2006.

[21] I. Stojmenovic, D. Liu, and X. Jia. "A scalable quorum based location service in ad hoc and sensor networks". *International Journal of Communication Networks and Distributed Systems*, 2007. To appear. http://www.site.uottawa.ca/~ivan/quorum-IJCNDS-F-prev.pdf.

[22] J. B. Tchakarov and N. H. Vaidya. "Efficient Content Location in Wireless Ad Hoc Networks". In *Proceedings of the 5th IEEE International Conference on Mobile Data Management (MDM)*, pages 74–85, 2004.

[23] G. Wang, G. Cao, and T. L. Porta. "Proxy-Based Sensor Deployment for Mobile Sensor Networks". In *Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, pages 493– 502, 2004.

APPENDIX
EXPERIMENTAL RESULTS FOR $0.1\% \leq PSP \leq 1\%$

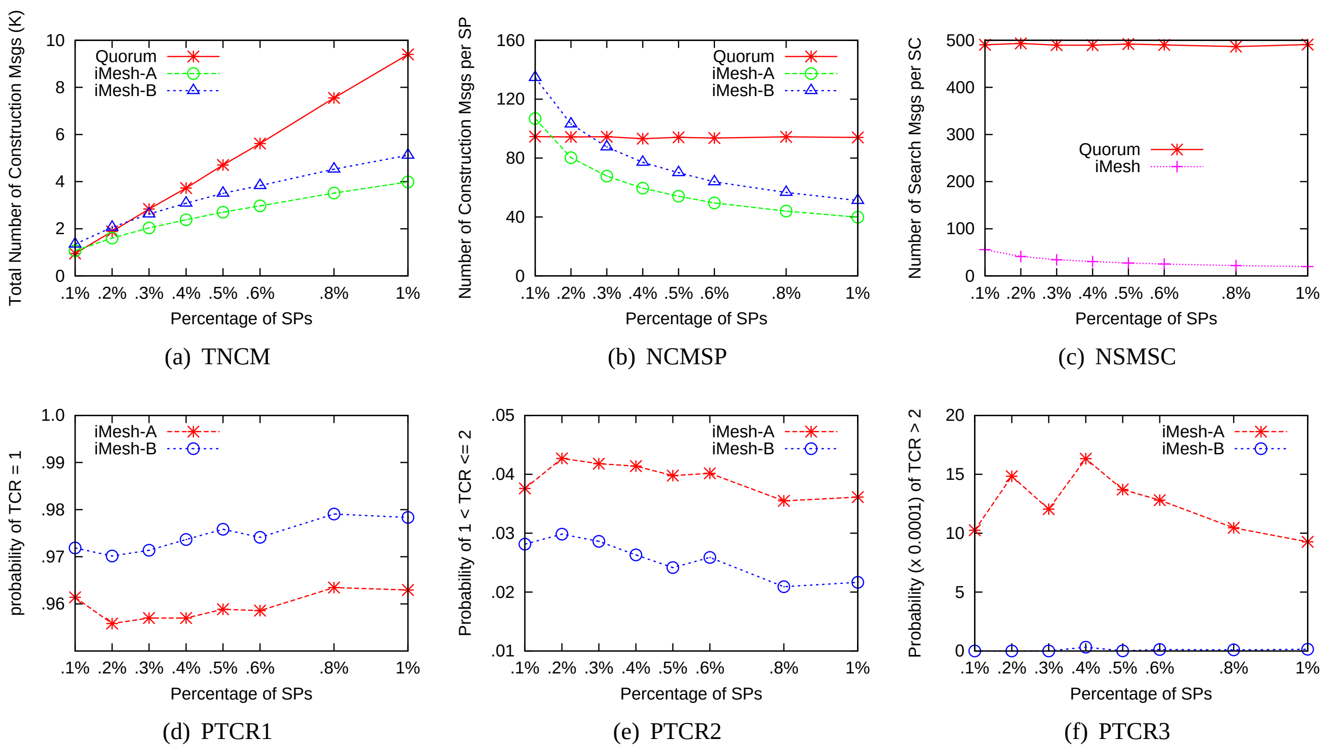


Fig. 12. Experimental results for $0.1\% \leq PSP \leq 1\%$