무선 센서 네트워크에서의 궤도 기반 콘텐츠 발간 및 구독을 위한 질의 이탈 방지

(Query Slipping Prevention for Trajectory-based Contents Publishing and Subscribing in Wireless Sensor Networks)

차 영 환 †
(Yeong Hwan Tscha)

요 약 본 논문은 무선 센서 네트워크에 있어서 궤도 기반 매취메이킹 서비스를 위한 질의 이탈 및 이의 방지에 관한 것이다. 이러한 문제는 정보 구독 궤도를 따라 전파되는 질의가 정보 발간 궤도와 기하학적으로는 겹침에도 불구하고 정보를 획득하지 못할 때 발생한다. 이에 따라 질의를 재 제출하거나 새로운 구독 궤도를 시작함으로 인한 시간 지연이 초래되어 최악에는, 궤도내의 루핑이나 네트워크 전체로의 메시지 범람을 야기한다. 이 문제를 정형적으로 다루고 그 해결책을 제시한다. 먼저, 노드들이 존재하는 영역을 논리적으로 작은 그리드들로 분할하고, 그리드 기반 멀티캐스트 다음-홉 선택 알고리즘을 제안한다. 제안 알고리즘은 궤도 설정을 직선형태로 유지하도록 시도함은 물론 수신 노드들의 분포 및 틈새 없는 그리드 단위의 멀티캐스트를 고려한다. 이러한 알고리즘에 의거하여 정보 발간 및 구독을 시행하는 경우 질의 이탈이 궁극적으로 방지됨을 증명한다. 제안 알고리즘이 탐욕적 송출과 같은 비 그리드 기반 알고리즘과 GAF와 같은 고정 크기의 그리드 접근법 보다 이웃 노드들의 전력을 더 적게 소모함을 알 수 있다.

키워드 : 센서 네트워크, 궤도 기반 매취메이킹 서비스, 콘텐츠 발간/구독, 다음-홉 선택 알고리즘, 질의 이탈, 전력 절약

Abstract This paper is concerned with the query slipping and its prevention for trajectory-based matchmaking service in wireless sensor networks. The problem happens when a query propagating along a subscribe trajectory moves through a publish trajectory without obtaining desired information, even though two trajectories intersect geometrically. There follows resubmission of the query or initiation of another subscribe trajectory. Thus, query slipping results in considerable time delay and in the worst, looping in the trajectory or query flooding the network. We address the problem formally and suggest a solution. First, the area where nodes are distributed is logically partitioned into smaller grids, and a grid-based multicast next-hop selection algorithm is proposed. Our algorithm not only attempts to make the trajectory straight but also considers the nodal density of recipient nodes and the seamless grid-by-grid multicast. We prove that the publishing and subscribing using the algorithm eventually eliminate the possibility of the slipping. It turns out that our algorithm dissipates significantly less power of neighbor nodes, compared to the non grid-based method, as greedy forwarding, and the fixed-sized grid approach, as GAF (Geographical Adaptive Fidelity)

Key words : Sensor networks, Trajectory-based matchmaking service, Contents publishing/subscribing, Next-hop selection algorithm, Query slipping, Power saving

1. Introduction

Portable computers and wireless technologies make feasible the implementation of small-sized intelligent sensor nodes[1]. We now witness a rush toward wireless sensor networks of several hundreds or thousands of nodes[2-4]. The networks are usually characterized by high nodal density, functionally/physically duplicated nodes, and infrastructure-less architectures. Power saving is a must in such networks, as they will be deployed using battery-powered nodes.

· 이 논문은 2004년도 상지대학교 안식년 지원으로 수행되었음
† 정 회 원 : 상지대학교 컴퓨터정보공학부 교수
yhtscha@mail.sangji.ac.kr
논문접수 : 2004년 9월 24일
심사완료 : 2005년 4월 8일

One of fundamental issues on wireless sensor networks is efficient message dissemination[3,5]. The quickly evolving *trajectory* paradigm [6-9] is worthy to receive attention. A source or server *publishes* or *updates*, perhaps periodically, information contents or available service list along a path called publish trajectory, while a sink or destination attempts to access it by propagating a *query* about its interests through a subscribe trajectory. The trajectory usually progresses in some *direction* without any particular destination in mind. *Matchmaking* takes place at the nodes where two trajectories intersect in common. In contrast to the strictquorum system[10], the paradigm does *not* require the explicit membership management and *decouples* the trajectory itself from the address-centric nature of the conventional path. This is very important because nodes in wireless networks may go into doze-mode, fail or move occasionally. Location service of routing protocols[8,11,12], contents listing[6,9,13], and data acquisition/disseminations [3,14,15] in the ad-hoc or sensor networks are typical examples which can be provided by the paradigm.

This paper is concerned with one fundamental issue on such trajectory-based paradigm. We particularly want to consider the question: *can matchmaking between publish and subscribe trajectories in the wireless networks be equally regarded as intersection of two line segments in Euclidian plane*? To state the problem formally, we begin with Definition 1.

- *Definition 1*: Given a trajectory T, nodes associated with it are as follows: 1) a node v explicitly chosen by some next-hop selection, i.e. packet forwarding algorithm, is said to be *repository node*; 2) a node u which is not repository but adjacent to a repository node v is said to be *quasi-repository* if it possesses publish/subscribe information, for instance, by overhearing advertising/subscribing packets to and from the repository nodes or other messaging with the repository nodes; 3) repository node v and quasi-repository node u are *quorums*[1]; 4) any non-quorum node w that resides in a cell defined by a repository node v is *quorum hole.*

- *Definition 2*: *Query slipping occurs* if some query along a subscribe trajectory T_S moves through quorum holes made by a publish trajectory T_P without obtaining its desired information, provided that T_P and T_S intersect geometrically.

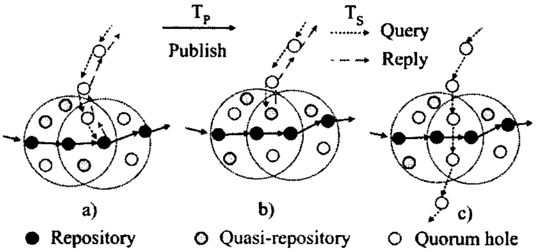


Fig. 1 Matchmaking and query slipping

There are three cases, as in Fig. 1, when T_P and T_S intersect. Matchmaking succeeds in both cases of a) and b), while it does not for case c). In the latter, it usually follows resubmission of the query message or initiation of another subscribe trajectory. Thus, query slipping may result in unnecessary time delay and in the worst, looping within the trajectory or query flooding over the network. This motivates us to study the query slipping and its prevention in the wireless sensor network.

We note that the primary criterion for the trajectory paradigm is not the least number of hops required for the trajectory development but seamless storing/propagation of update and query messages along the trajectory. The next-hop node on the trajectory is usually chosen by an algorithm that seeks the *least* number of hops, i.e. the shortest distance to some destination, as greedy forwarding[16] or MFR (Most Forward within Radius)[17]. Some recent works[6,8,9] related to the trajectory paradigm are based on such shortest-path principle. The trajectory constructed by the method has it that the physical distance between two consecutive nodes on the trajectory is *relatively* long hence, the adjacent nodes to them

---

[1] Similar to pseudo-quorum[6], in this paper, the term "quorum" means any network node that has some advertising or subscribing message required for the trajectorybased matchmaking.

may remain untouched when the advertising messages propagate alongthe trajectory. In later, some query event may move through these untouched nodes. Thereby matchmaking fails without obtaining desired information. Put differently, even though two trajectories intersect geometrically[2], there exists a *non-zero* possibility that they do not possess commonly interesting nodes in the overlapped space. This motivates us to study the query slipping and its prevention in the wireless sensor network. Neither reliable 1-hop *broadcasting* for publishing nor *polling* for subscribing will be sufficient, as each is accompanied with many messages and considerable time-delay.

The rest of this paper is as follows. Related works are reviewed in the next section. Section 3 defines the network model that we adopt. Grid index is introduced to embed various-sized grids within a cell. Section 4 concentrates on a gridbased next-hop selection algorithm that exploits a geographic grid-by-grid multicast. In Section 5, we prove the next-hop algorithm eventually prevent the query slipping problem in the wireless sensor networks. The hop-distance and the power-saving ratios are given for the performance evaluation. Finally, the paper concludes with Section 6.

2. Overview of Related Work

The heart of the trajectory-paradigm is the next-hop selection algorithm. Niculescu and Nath [7] study various topologies and next-hop selection methods: the closest node to the trajectory, the node given by MFR, the centroid of feasible candidates, the node with the most battery left, and a random node. Other recent works[6,8,9], nevertheless, prefer a simple one, called cross-shaped trajectory or *column/row* method.

Aydin and Shen[6] select the farthest neighbor from the current node as the next-hop in order to forward the publish/subscribe packet at the least cost. Alternatively, to keep the trajectory close to the straight line in the desired direction, they also consider the node closest to the line. Tchakarov and Vaidya[9] attempt to attain two goals together. The node farther away and closest to the line is determined by finding the node that has the largest value of the rating function d/r where, d is the distance from the current node and r is the offset from the line segment in the direction. Luo, Hubaux, and Eugste[13] propose a gossip-based multicast protocol in order to improve the robustness of the quorum access. All researchers try to show how each method effectively contributes the success rate of the query messages.

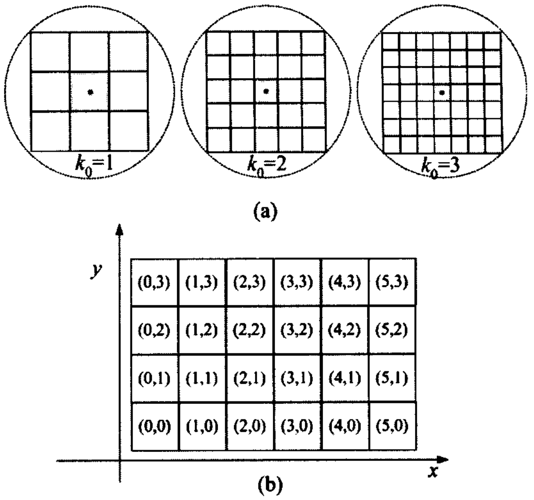
Aydin and Shen[6] and Tchakarov and Vaidya [9] also mention the effect of overhearing nodes, i.e. quasi-repository nodes by Definition 1 in their simulation. However, neither an explicit description of such node's actions nor the query slipping is explored. Their works appear to simply take into account the broadcast nature of the wireless medium. Furthermore, overhearing itself is not reliable hence, we are not sure all neighbors receive publish/subscribe packets correctly or not. The hidden or exposed terminals lower the reliability as well. The uncertainty may be relieved by the p-thickness idea of Stojmenovic[8]: the packets traveling along the trajectory further propagate to the p-hop outward nodes. However, this scoped flooding generates many packets and dissipates unnecessarily the power of neighbors.

Xu, Heidmann, and Estrin[19] propose the *virtual grid* concept for dense ad hoc networks such that, for adjacent grids A and B, all nodes in A can communicate with all nodes in B and vice versa. One specific node per grid takes the responsibility of routing, while remaining nodes go periodically to the doze-mode for power-saving. The network life time is directly proportional to the node density per grid. Hence, *more than one* node per grid is meaningful in this study. In summary, the query slipping problem and its solution is still open problem. The virtual grid terminology we use coincides with that as in [19], but its definition and meaning are different (see Section 3). As opposed

---

[2] Given two straight line segments respectively connecting each pair of two nodes given by their positions, i.e., (x,y)-coordinate values, we can easily verify whether such segments geometrically intersect or not in Euclidian plane using some well-known algorithm in computational geometry area(see [18], for instance).

to their work, our approach does not require a gateway or leader in order to route messages. Furthermore, our next-hop selection algorithm can be incorporated into any existing trajectory setup protocols such as [6,8,9,19].



(0,3)	(1,3)	(2,3)	(3,3)	(4,3)	(5,3)
(0,2)	(1,2)	(2,2)	(3,2)	(4,2)	(5,2)
(0,1)	(1,1)	(2,1)	(3,1)	(4,1)	(5,1)
(0,0)	(1,0)	(2,0)	(3,0)	(4,0)	(5,0)

Fig. 2 Virtual Grid Model

3. Virtual Grid Model

As shown in Fig. 2 a), we logically partition each radio cell of radius R(m) into $(2k_0+1)\times(2k_0+1)$ virtual grids according to Eqn. (1) where, each grid is a square of $r\times r$(m) and $k_0\geq 0$.

$$(k_0r+r/2)^2 + (k_0r+r/2)^2 \leq R^2 \qquad (1)$$

• *Definition* 3: Denote by *g(p,q)* a grid whose *(x,y)*-coordinate value is *(p,q)* in Euclidian plane represented by virtual grids as Fig. 2 b). Given *g(s,t)* and *g(e,f)*, let $d = max\{|s-e|_a, |t-f|_a\}$ where, $s,e,t,f\geq 0$, $|A|_a$ denotes the absolute value of integer *A*, and $\max\{U,V\}$ means *U* if $U\geq V$ and *V* otherwise. Then, we say that *g(s,t)* and *g(e,f)* are *d*-distanced each other.

Fig. 2 a) shows embedding of various-sized grids within a cell where, k_0=1,2,3. We call k_0 *grid index*, as it determines the size of the grid and finally, the number of grids per cell. Note that the grid index is a tunable parameter. For instance, it may be used to create many small grids for the dense networks or a few large grids for the sparse networks. It also closely relates to the length of a nominal radius R of a cell. We assume that, using some link-level protocol, a node exchanges its grid id with its 1-hop neighbors, i.e. up to k_0-distanced nodes. For instance, grids *g(2,2)* and *g(5,1)* are 3-distanced and are neighbors for k_0=3, but not neighbors for k_0=2. Throughout this paper, we consider that the cross-shaped trajectory [6-9] for its simplicity and favorable success rate in matchmaking and that the trajectory always makes it possible to progress toward four directions: East, West, North, and South. This implies that there exists at least one node in $k_0\times(2k_0+1)$(=rows×columns) grids laid in the direction of the trajectory progress.

Even though the grid index k_0 is tunable at the network level, it solely can not accommodate the dynamic situation such as node failure or congestion in the sensor networks. Depending on the nodal-density, the recipient nodes may also distribute over several grids. Thereby, we deploy multicasting for the development of the trajectory. For the simplicity, we name not only the grids that hold the target nodes, but also their some neighbor grids, in order for the whole recipient grids to be *rectangular*. The set of recipients is then represented by two coordinate values within it, the minimum and the maximum. For instance, if a set of recipient grids is given as <(1,2),(3,4)> then, it identifies the rectangle of encompassing the four corner grids g(1,2), g(1,4), g(3,2) and g(3,4): a total of nine grids. Note that we use *grid-centric* rather than *node-centric* naming, even though we may still need to pinpoint some specific nodes by their addresses in some cases.

4. The Proposed Next-Hop Selection Method

4.1 Nodal Density-Aware Grid-Based Selection

Considering the density of the sensor networks, we ignore the position of the originating node and try to keep the trajectory close to the straight line inthe desired direction, as in [6,9]. However, the nodal density-awareness and grid-based multicasting are exploited for the sensor networks to be free from the query slipping. Careful consideration is also made to guarantee the seamless storing/propagating of the publishing/subscribing messages.

• *Definition 4*: Suppose that a node *s* in a grid

$g(x_s,y_s)$ develops its trajectory in some direction d. Then, a set of k_0 grids which intersect the line segment to d from s is said to be *default repository set* G_s^d such that $G_s^{East} = \{g(x_s+1,y_s), g(x_s+2,y_s), \cdots, g(x_s+k_0,y_s)\}$, $G_s^{West} = \{g(x_s-1,y_s), g(x_s-2,y_s), \cdots, g(x_s+k_0,y_s)\}$, $G_s^{North} = \{g(x_s,y_s+1), g(x_s,y_s+2), \cdots, g(x_s,y_s+k_0)\}$, $G_s^{South} = \{g(x_s,y_s-1), g(x_s,y_s-2), \cdots, g(x_s,y_s-k_0)\}$ for each direction respectively, where k_0 is the grid index.

Let M_s^g denote the set of grids, chosen by a node s, where recipient nodes, i.e. the set of next-hop nodes reside. Let $|M_s^g|$ be the number of the nodes in M_s^g. The default repository set G_s^d lends itself to be mandatory in Msg because all the grids in G_s^d intersect the line in the desired direction. Denote by N_0 the minimal number of the next-hop nodes required per hop in order to develop a trajectory. If $|M_s^g| \geq N_0$ then, the shortest distance to the next-hop could be made. However, this would be insufficient because any of grids in M_s^g may be empty, as it has no nodes in it. To cope with the requirement, we introduce the definition of i-distance left/right set, as follows.

- *Definition 5*: Given G_s^d, its *i-distance left/right set* $G_s^d(i,l)/G_s^d(i,r)$ is given as follows for each direction: $G_s^{East}(i,l)/G_s^{East}(i,r) = \{g(x_s+1,y_s+/-i), g(x_s+2,y_s+/-i), \cdots, g(x_s+k_0,y_s+/-i)\}$, $G_s^{West}(i,l)/G_s^{West}(i,r) = \{g(x_s-1,y_s-/+i), g(x_s-2,y_s-/+i), \cdots, g(x_s-k_0,y_s-/+i)\}$, $G_s^{Noth}(i,l)/G_s^{North}(i,r) = \{g(x_s-/+i,y_s+1), g(x_s-/+i,y_s+2), \cdots, g(x_s-/+i,y_s+k_0)\}$, $G_s^{South}(i,l)/G_s^{South}(i,r) = \{g(x_s+/-i,y_s-1), g(x_s+/-i,y_s-2), \cdots, g(x_s+/-i,y_s-k_0)\}$ where, $i \in \{1, 2, .., k_0\}$.

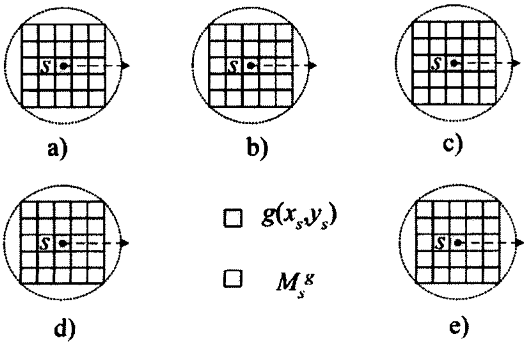


Fig. 3 Grid-based multicast next-hop selection using i-distance left/right set

Given Fig. 3 a), if either $g(x_s+1,y_s)$ or $g(x_s+2,y_s)$ in the default repository set, G_s^{East}, is empty then, its 1-distance left node set, $G_s^{East}(i,l)$ $(=\{g(x_s+1,y_s+1), g(x_s+2,y_s+1)\})$, is firstly added to M_s^g as in ·b). Now, if $|M_s^g| \geq N_0$ and each row in M_s^g has at least one node then, all nodes in M_s^g become the next-hop nodes. If not, we add the 1-distance right set, $G_s^{East}(i,r)$ $(=\{g(x_s+1,y_s-1), g(x_s+2,y_s-1)\})$, as shown in c). If the updated M_s^g still does not satisfy the termination

```
// Algorithm GBS: Grid-Based Selection of the Next-Hop
// Input: $N_0$, $k_0$ // Output: $M_s^g$, $n_s$

        // assume the trajectory makes a progress toward the east
line 1:  $min_x = max_x = x_s + 1$; $min_y = max_y = y_s$;
// $C_s^g$ : <$(min_x, min_y), (max_x, max_y)$>, the name of $M_s^g$.
         // $ns_y$ is the y-coordinate value of the grid $n_s$ resides in
line 2:  $M_s^g = \phi$, $G_s^e = \phi$, $C_s^g = \phi$, $n_s = \phi$, $ns_y = y_0$;
line 3:  $M_s^g = G_s^{East} = \{g(x_s+1, y_s), \ldots, g(x_s+k_0, y_s)\}$;
line 4:  for $j$ = 1 to $k_0$  // find node-empty repository grids
line 5:    if $|g(x_s+i, y_s)| == 0$ then $G_s^e = G_s^e \cup g(x_s+i, y_s)$;
line 6:    else $max_x = x_s + i$;
line 7:  if ($|M_s^g| > N_0$) and ($G_s^e == \phi$) then {
line 8:    $C_s^g$ = <$(min_x, min_y), (max_x, max_y)$>;
           // farther from $s$ and closest to the line
line 9:    let $n_s$ be an arbitrary node in $g(max_x, ns_y)$;
line 10:   exit;
line 11: }
        // find $M_s^g$, the set of the next-hop nodes
line 12: $i$=1;
line 13: for each $g(x_s+t, y_s) \in G_s^e$, $t \in \{1,2, \ldots, k_0\}$ {
line 14:   while ($i < k_0+1$) {
line 15:     if $|g(x_s+t, y_s+i)| > 0$ then {  // $i$-distance left set
line 16:       $M_s^g = M_s^g \cup G_s^{East}(i,l)$;
line 17:       $max_y = y_s + i$; $G_s^e = G_s^e - g(x_s+t, y_s)$;
line 18:         for each $g(x_s+t', y_s) \in G_s^e$, $t' \in \{1,2, \ldots, k_0\}$ {
line 19:           if $|g(x_s+t' y_s+i)| > 0$ then { $G_s^e = G_s^e - g(x_s+t' y_s)$;
line 20:           if ($x_s+t' > max_x$) then $max_x = x_s + t'$;
line 21:           }
line 22:         }
line 23:         break;
line 24:       } // end of if-statement on line 15 and
line 25:     else if $|g(x_s+t, y_s-i)| > 0$ then {  // $i$-distance right set
line 26:       $M_s^g = M_s^g \cup G_s^{East}(i,r)$;
line 27:       $min_y = y_s - i$; $G_s^e = G_s^e - g(x_s+t, y_s)$;
line 28:         for each $g(x_s+t', y_v) \in G_s^e$, $t' \in \{1,2, \ldots, k_0\}$ {
line 29:           if $|g(x_s+t', y_s-i)| > 0$ then { $G_x^e = G_s^e - g(x_s+t' y_x)$;
line 30:            if ($x_s+t' > max_x$) then $max_x = x_s + t'$;
line 31:           }
line 32:         }
line 33:       break;
line 34:       } // end of else-if statement one line 25
line 35:     else
line 36:       $i = i+1$;
line 37:   } // end of while-statement on line 14
      // find $n_s$, the node farther and closer to the line
line 38:     if ($x_s+t > max_x$) then $max_x = x_s + t$;
line 39:     if ($|M_s^g| > N_0+1$) and ($G_s^e == \phi$) then {
line 40:     $C_s^g$ = <$(min_x, min_y), (max_x, max_y)$>;
line 41:       if $|g(x_s+k_0, y_s)| > 0$ then $ns_y = y_s$;
line 42:       else {  // find farther and closer to the line
line 43:       $ns_y = max_y$;
line 44:       for $i = min_y$ to $max_y$
line 45:         if ($g(max_x, i) > 0$) && ($|i-y_s|_a < |ns_s - y_s|_a$) then
line 46:           $ns_y = i$;
line 47:       }
line 48:     let $n_s$ be a node in $g(max_x, ns_y)$;
line 49:       exit;
line 50:   } // end of if-statement on line 39
line 51: $i$=1;
line 52: } // end of for-statement on line 13
```

Fig. 4 Pseudo code of GBS

condition set (see below), additional steps will follow until $i=k_0$(=2 in this example), as in d) and e). At last, nodes in the final M_s^g become the next-hop nodes of s, the current node. And one of them, farther away from s and closest to the line in the direction, is assigned to the backbone node ns. It is responsible for finding the succeeding next-hop nodes and multicasting the publishing and subscribing message. Fig. 4 shows a formal description of the algorithm, GBS(Grid-Based Selection of the next-hop nodes), discussed so far.

Termination condition set of the next-hop selection: 1) if $|M_s^g| \geq N_0$ and 2) either $i=k_0$ for $G_s^d(i,r)$ or every row in M_s^g contains at least one node.

- *Corollary 1*: For M_s^g, the followings hold. 1) $|M_s^g| \geq 1$. 2) Let there exist some $g(x_s+i,y_s)$ and $g(x_s+i,y_s \pm j)$ such that $g(x_s+i,y_s) \in G_s^e$ and $|g(x_s+i,y_s \pm j)|>0$, $1 \leq i,j \leq k_0$. Then, M_s^g always contains either some $g(x_s+i,y_s+h)$ such that $|g(x_s+i,y_s+h)|>0$ or $g(x_s+i,y_s-f)$ such that $|g(x_s+i,y_s-f)|>0$ where, $1 \leq f,h \leq j \leq k_0$.

Proof: 1) Self-evident by assumption on the trajectory development. 2) Since $g(x_s+i,y_s) \in G_s^e$, i-distance left/right set $G_s^d(i,l)/G_s^d(i,r)$ will be considered at most until $i=k_0$(lines 14-37 of GBS in Fig. 4). The property follows as there exits $g(x_s+i,y_s \pm j)$ such that $|g(x_s+i,y_s(j)|>0$ for $1 \leq i,j \leq k_0$.

- *Corollary 2*: M_s^g is found with $O(d)$ time regardless of the number of nodes in the network where, d is some constant.

Proof: The worst case time complexity is determined by the part corresponding to lines 13-37 where, for-while-for statements are respectively taken a, b and c times such that $a+b \leq k_0$ and $c \leq k_0$. Hence, we have the worst time complexity $O(k_0^3)$. But k_0 is a constant, the complexity becomes $O(d)$ letting $d = k_0^3$.

4.2 Trajectory Construction Example

Fig. 5 shows an example of the trajectory setup by the algorithm GBS. Given N_0=2, at least, two nodes per hop are required to find M_S^g, and the 2-distanced grids are reachable at most by each ns. Initiating node X is in $g(2,2)$ and the trajectory goes for the eastward thus, it is seen that $G_X^{East}=\{g(3,2),g(4,2)\}$. As all grids in G_X^{East} have nodes and $|G_X^{East}|(=4) > N_0(=2)$, the first hop is finished by assuming n_X=D, the farthest from X, even there are B, C, D in $g(4,2)$. Note that all the nodes in the default repository grid set become the next-hop nodes regardless of how many are in it. Similar argument is also applicable to the rest hops. For M, initially, $M_M^g = G_M^{East}$ (=$\{g(9,2),g(10,2)\}$), and $|M_M^g|=N_0(=2)$ but, $g(10,2)$ is empty. Hence, $G_M^{East}(1,1)$ (=$\{g(9,3),g(10,3)\}$) is added to M_M^g. In case of node P, $|G_P^{East}| = |\{g(11,3),g(12,3)\}| = 0$ thus, $G_P^{East}(1,l) = \{g(11,4),g(12,4)\}$, and $G_P^{East}(1,r) = \{g(11,2),g(12,2)\}$ is added. Making $M_P^g = \{g(11,2), g(12,2), g(11,3), g(12,3), g(11,4), g(12,4)\}$ and $|M_P^g|=2(=N_0)$, it follows $C_P^g = \langle(11,2),(12,4)\rangle$ and n_P = R, the destination.

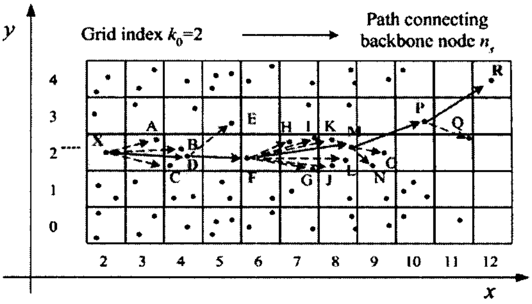


Fig. 5 Trajectory setup example by GBS

5. Correctness And Performance

5.1 Correctness

- *Theorem 1*: Assume that the publish trajectory T_P and the subscribe trajectory T_S, established using the algorithm GBS, intersect geometrically. Then, matchmaking always succeeds without query slipping.

Proof: As shown in Fig. 6, consider four nodes

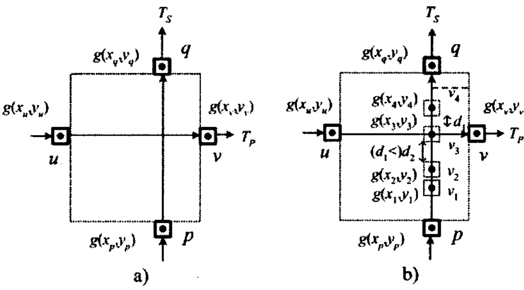


Fig. 6 Intersection of T_P and T_S

u, v, p and q, respectively, residing in grids $g(x_u,y_u)$, $g(x_v,y_v)$, $g(x_p,y_p)$, and $g(x_q,y_q)$ where, $x_u \le x_q(=x_p) \le x_v$, $y_p \le y_u(=y_v) \le y_q$. Let T_P be established from u to v in the direction of the eastward, while T_S is to be from p to q toward the north such that $x_u < x_v \le x_u+k_0$ and $y_p < y_q \le y_p+k_0$ where, k_0 is grid index. Let the overlapping space by both T_P and T_S correspond to the rectangle that comprises four corner grids $g(x_u+1,y_p+1)$, $g(x_u+1,y_q-1)$, $g(x_v-1,y_q-1)$, and $g(x_v-1,y_p+1)$. There are two cases: when the rectangle does not contain any node, as in Fig. 6 a) (case 1), and when it does, as in Fig. 6 b) (case 2). The proofs are based on contradiction: we show that the geometrical crossing without sharing any common node is impossible for each case. 1) Case 1: Noting the rectangle does not contain any node, $g(xq,y_q)$, $g(x_v,y_v)$, and $g(x_p,y_p)$ must be included in Mug by Corollary 1 and inequality equations, $x_u < x_v \le x_u+k_0$ and $y_p < y_q \le y_p+k_0$. This means that u always selects q, p and v as the next-hop nodes in case of T_P establishment. As p is a repository node possessing published contents, T_S from p to q is impossible. 2) Case 2: Suppose there exits some node v_1 in the rectangle such that it has not received the contents advertised by v while has got the query event from p. For v_1 to be such quorum hole node, there must exist other nodes along the same x-coordinate as v_1. In Fig. 6 b), three nodes, v_2 in $g(x_2,y_2)$, v_3 in $g(x_3,y_3)$, and v_4 in $g(x_4,y_4)$ are depicted where, $y_p+1 \le y_1 < y_2 < y_3 < y_4 < y_q$, $d_1(=|y_4-y_u|a) < d_2(=|y_2-y_u|a)$, $y_u=y_3=y_v$, and $x_1=x_2=x_3=x_4=x_p=x_q$. As $y_p < y_q \le y_p+k_0$, $\{g(x_i,y_i) | 1 \le i \le 4\} \in G_p^{North}$ $(\subseteq M_p^g)$ by Corollary 1 where, G_p^{North} is the default repository set chosen by p. We know $g(x_p,y_3)$ where v_3 resides also belongs to G_u^{East} as $y_3=y_u$. Thus, v_3 is the common node of the two trajectories. If such v_3 dose not exist then, v_4 will be as $d_2>d_1$. Unless v_4 exists, then v_2 does. If v_2 does not, then v_1 will be. But,the last case violates the assumption that v_1 is not shared by both trajectories. This is contradiction. If v_1 does not exist then, case 2 becomes case 1 above. With similar argument we can easily prove for the case $y_u \neq y_v$ and $x_q \neq x_p$.

5.2 Experimental Performance Results

Consider the nodes which are residing in the grids under the radio umbrella defined by each backbone node ns, but not chosen as the next-hop in the trajectory. And let them go periodically to sleeping state for power-saving[3]. Assuming the uniform distribution of nodes, a server or client that deploys our next-hop selection algorithm GBS for the establishment of the cross-shaped trajectory occupies $4k_0+1$ grids in the best case and $(2k_0+1)\times(2k_0+1)$ grids in the worst. Non grid-based protocols, such as greedy forwarding [16] or MFR [17], always cover the area of πR^2, i.e. slightly wider than the area of $(2k_0+1)\times(2k_0+1)$, where $R=[(2k_0+1)r/\sqrt{2}]$ by Eqn. (1). Let A be the area corresponding to the set of grids per cell used by GBS. Denote by B, πR^2, the area covered by the non-grid based forwarding method. The ratio given by Eqn. (2) below is said to be *power-saving neighbor (node) ratio for a server (or client)* because once the trajectory has been set up the only nodes that reside in the selected grids are concerned with subsequent publishing and subscribing.

$$1 - (A/B) \qquad (2)$$

To validate and measure the power-saving neighbor ratio of a server or client, we wrote a simulation program that calculates various power-saving ratios under different combinations of grid index (k_0), nodal density per grid (ρ), and number of next-hop nodes (N_0). We assumed that the network is dense and static and that nodes are uniformly distributed over the network. For the simplicity we also did not consider the irregularities of the radio cells and grids. As stated in Section 3, we assumed thatthe trajectory always makes it possible to progress toward four directions: East, West, North, and South. This means that there exists at least one node in $k_0\times(2k_0+1)$ grids laid in the direction of the trajectory progress.

We first vary k_0 from 1 to 4, while using $\rho=0.5$. Fig. 7 shows the power-saving neighbor ratios of a server (or client) initiating a cross-shaped trajectory, when N_0 is given as 1, 2, 3, 4, respectively. It is seen that the ratios generally

---

[3] We do not want to repeat the state-transition diagram or procedure for such a typical power-saving mode. Readers may refer to [20] for more detailed information.

increase as k_0 increases and N_0 decreases, and that they fall between the theoretical bounds from 36% to 87%. When $k_0<4$, the saving ratios are not so sensitively depending upon N_0. However, the ratios degrade as N_0 increases. Fig. 8 shows the power-saving neighbor ratios when ρ varies in the range [0.5, 0.7, 0.9, 1.0], as functions of k_0, while fixing $N_0=3$. As expected, the savings become more apparent as k_0 increases. Interestingly, in case of $\rho<1.0$, the ratios are invariants for each given k_0. This is because the number of grids considered by the algorithm GBS is given as a multiple of roughly $4k_0$, so it can tolerates many cases in which $\rho<1.0$ and N_0 is relatively small as this case of 3.

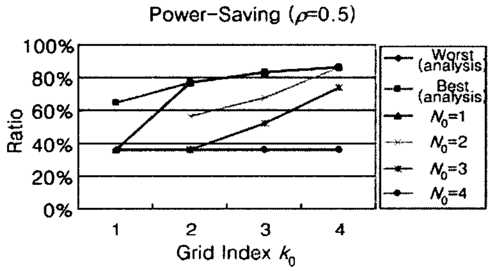


Fig. 7 Power-saving neighbor ratios as functions of k_0 and N_0

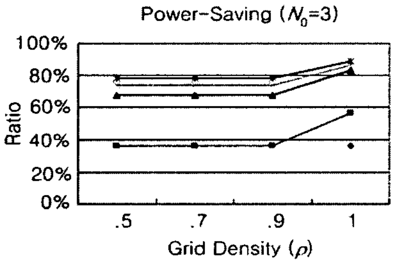


Fig. 8 Power-saving neighbor ratios as functions of ρ and k_0.

One may argue that our approach is only viable for unrealistic small N_0 (say, 2 or 3) or high ρ (say 1.0) or large k_0 (say, 5 or 6). The story, however, is different since, as also shown in [2], small N_0 (say, 2 or 3) is enough to reliably multicast a message, unlike a one-to-one forwarding (i.e. $N_0=1$). Achieving $\rho\approx1.0$ is easy by tuning k_0 and N_0 such that $k_0\approx N_0$. Note that, as shown in Fig. 7 and Fig. 8, our approach still allows neighbor nodes to save their power even for the case of $\rho<1.0$. This is quite different from the previous fixed-grid based topology control approach for the power-saving, as GAF(Geographical Adaptive Fidelity)[19], which is required to have more than two nodes per grid for the power-saving to be attained. Meanwhile, the radius R(m) of the cell varies from 10 to 50 in the sensor networks [6-8,12] and 250 in the ad hoc networks [19] thus, we expect k_0 is not lager than 4 or 5. And N_0 may not be so large, too. This is because, in terms of the trajectory reliability, for instance, $N_0=2$ or 3, nearly identical to k_0, may suffice. Thus our approach could be viable for reality[4]. Finding optimal value of k_0 is as follows.

• *Corollary 3*: The algorithm GBS uses the minimal number of grids per cell with the highest power-saving, if k_0 and N_0 are such that $N_0\le k_0=\lfloor(\sqrt{n}-1)/2\rfloor$ where, n is the number of nodes uniformly distributed over $(2k_0+1)\times(2k_0+1)$ grids,

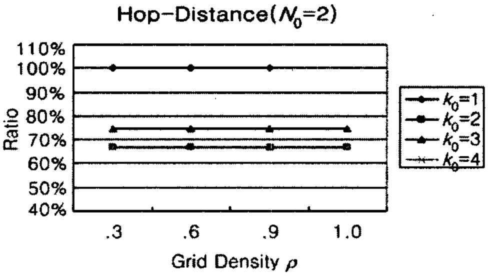


Fig. 9 Hop-distance ratios as functions of ρ and k_0

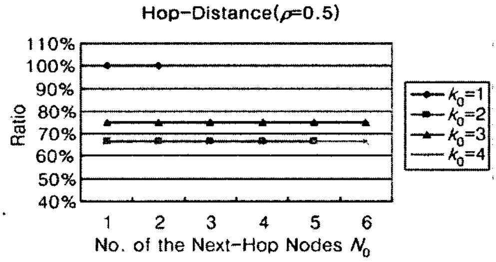


Fig. 10 Hop-distance ratios as functions of N_0 and k_0

---

[4] Unlike a server or client that initiates its trajectory into four directions, a 'transit' backbone node which simply relays a publishing or subscribing message from one neighbor to another neighbors certainly dissipates less power of neighbors, as its trajectory is set up in the direction of a column(or row)-wise. In this paper, however, we have not covered this subject.

and $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x.

Proof: The density per grid ρ is given as $\rho = n/(2k_0+1)^2$. If ρ=1 and $N_0 \leq k_0$, then the minimal number of grids, $4k_0$+1, per cell will suffice. Keeping k_0 maximal and letting $n/(2k_0+1)^2$ = 1, it is given that k_0 becomes $\lfloor (\sqrt{n}-1)/2 \rfloor$.

The trajectory usually makes it progress in some direction as created by without any particular destination in mind. The conventional *hop-count* metric, as in routing algorithms, *may not be useful for* the valuation of the trajectory-based paradigm. Thus, we consider the *average hop-distance* between two consecutive backbone nodes in the trajectory. Throughout our simulations, we assumed that a node always placed at the center of a grid, while keeping $0<\rho<1$, and measured the hop-distance as the number of grids between two repository nodes. We further normalized the distance with the number of grids corresponding to the cell radius R.

Fig. 9 and Fig. 10 show hop-count ratios, respectively, as functions of ρ, N_0 and k_0. All ratios except for k_0=1 are approximately 70% of the cell radius R. It turns out that the ratios are almost invariant with ρ and N, while depending on k_0, but the deviations are still insignificant as k_0 increases. The characteristic of power-saving of neighbor nodes comes with a shortcoming of such relatively short distance between backbone nodes on the trajectory. The hop-distance ratio can be explained as follows. Given ρ=1, the trajectory built by the GBS always makes it go straight. By Eqn. (1) in Section 3, the longest distance from the center in a cell is R (=$[(2k_0+1)^2 r^2]/2$), while the allowed in our approach is $k_0 r+(r/2)$. Comparing these, it follows that $[k_0 r+(r/2)]/R$ = $1/\sqrt{2}$(≈0.71). However, in the case of $0<\rho<1$, if the trajectory moves in the 'zig-zag' style toward the diagonal direction, then it will span R at most.

6. Conclusions

In this paper, we studied the query slipping problem and its prevention for the trajectory-based matchmaking paradigm in wireless sensornetworks. We addressed the issue by logically dividing the space of the whole network into smaller grids andproposed a new next-hop selection algorithm by taking advantage of the seamlessly grid-by-grid multicasting. It is noticeable that the primary criterion for the trajectory paradigm is not the least number of hops required for the trajectory setup but seamless storing/propagation of update and query messagesalong the trajectory. Our analysis and simulation show that our algorithm less dissipates the power of 1-hop neighbors. Compared to the non-grid-based protocol, as greedy forwarding, from 36 % to 87% of neighbor nodes can go to sleeping state for power-saving. It is given thatthe average hop-distance ratios of the proposed algorithm are approximately 70% of a cell radius.

The study on the impact of the proposed algorithm underirregularity ands is left for further study. In-depth works on the query slipping problem considering node mobility, sparse network, and non-uniform distribution of nodes also deserve to receive considerable attention.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, Vol. 40, No. 8, 2002, pp. 102-114.

[2] B. Deb, S. Bhatnagar and B. Nath, "Information assurance in sensor networks," *Proc. WSNA'03*, pp. 160-168.

[3] C. Intanagonwiwat, R. Govindran, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," *Proc. MobiCom'00*, pp. 56-67.

[4] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Mobile networking for smart dust," *Proc. MobiCom'99*, pp. 271-278.

[5] B. Krishanamachari, D. Estrin, and S. Wicker, Modelling data-centric routing in wireless sensor networks, USC Computer Engineering Technical Report CENG 02-14, 2002.

[6] I. Aydin, C.-C. Shen, "Facilitating match-making service in ad hoc and sensor networks using pseudo quorum," *Proc. ICCCN'02*, pp. 14-19.

[7] D. Niculescu and B. Nath, "Trajectory based forwarding and its applications," *Proc. MobiCom'03*, pp. 260-272.

[8] I. Stojmenovic, A scalable quorum based location

update scheme for routing in ad hoc wireless networks, SITE TR-99-09, University of Ottawa, 1999.

[9] J. B. Tchakarov and N. H. Vaidya, "Efficient content location in wireless ad hoc networks," *Proc. MDM'04*, pp. 74-85.

[10] Z. J. Hass and B. Liang, "Ad hoc mobility management with uniform quorum systems," *IEEE/ACM Transactions on Networking*, Vol. 7, No. 2, 1999, pp. 228-240.

[11] T. Camp, Location information services in mobile ad hoc networks, TR MCS-03-15, The Colorado School of Mines, 2003.

[12] S. Giordano, I. Stojmenovic, and L. Blazevic, Position based routing algorithms for ad hoc networks: a taxonomy, in *Ad Hoc Wireless Networking*, Amsterdam: Kluwer, 2003.

[13] J. Luo, J.-P. Hubaux, and P. Eugster, "PAN: providing reliable storage in mobile ad hoc networks with probabilistic quorum systems," *Proc. MobiHoc'03*, pp. 1-12.

[14] D. Braginsky and D. Estrin, "Rumor routing algorithm for sensor networks," *Proc. WSNA'02*, pp. 22-31.

[15] A. Cerpa, J. Elson, M. Hamilton, and J. Zhao, "Habitat monitoring: application driver for wireless communication technology," *Proc. 2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean* , pp. 20-34.

[16] B. Karp and H. T. Kung, "Greedy perimeter stateless forwarding for wireless networks," *Proc. MobiCom'00*, pp. 243-254.

[17] T.-C. Hou and V. Li, "Transmission range control in multihop packet radio networks," *IEEE Transactions on Communications*, Vol. 34, No. 1, 1986, pp. 38-44.

[18] F. P. Preparata and M. I. Shamos, Computational geometry: an introduction, New York: Springer-Verlag, 1985.

[19] Y. Xu, J. Heidemann and D. Estrin, "Geography-informed energy conservation for ad-hoc routing," *Proc MobiCom'01*, pp. 70-84.

차 영 환

1979년 3월~1983년 2월 인하대학교 전자계산학과(학사). 1983년 3월~1985녀 2월 한국과학기술원 전산학과(석사) 1989년 9월~1993년 2월 인하대학교 대학원 전자계산학과(박사). 1985년 3월~1990년 2월 한국전자통신연구원 선임연구원
1986년 3월~1997년 2월 미국 NIST(NBS) 객원과학자
2004년 3월~2005년 2월 터어키 Bogazici Univ. 객원교수
1994년 3월~현재 상지대학교 전임강사, 조교수 및 부교수
관심분야는 네트워크 구조, 통신 프로토콜