

Article

Adaptive Square-Shaped Trajectory-Based Service Location Protocol in Wireless Sensor Networks

Hwa-Jung Lim, Joa-Hyoung Lee and Heon-Guil Lee *

Dept. of Computer Science and Engineering, Kangwon National University, Chuncheon Gangwondo, 200-701, Korea; E-Mails: jinnie4u@kangwon.ac.kr (J.-H.L.); hjlim@kangwon.ac.kr (H.-J.L.)

* Author to whom correspondence should be addressed; E-Mail: hglee@kangwon.ac.kr; Tel.: +82-01-4914-0107; Fax: +82-33-252-6390.

Received: 22 March 2010; in revised form: 8 April 2010 / Accepted: 14 April 2010 /

Published: 30 April 2010

Abstract: In this paper we propose an adaptive square-shaped trajectory (ASST)-based service location method to ensure load scalability in wireless sensor networks. This first establishes a square-shaped trajectory over the nodes that surround a target point computed by the hash function and any user can access it, using the hash. Both the width and the size of the trajectory are dynamically adjustable, depending on the number of queries made to the service information on the trajectory. The number of sensor nodes on the trajectory varies in proportion to the changing trajectory shape, allowing high loads to be distributed around the hot spot area.

Keywords: service location; trajectory; data replication; load scalability; robust

1. Introduction

Advances in wireless networking have set new paradigms in computing, including pervasive computing based on a large-scale wireless sensor network. A wireless sensor network, a type of *ad hoc* network (MANets), is designed to be an infrastructure-less, unattended, and rapidly-deployable network. A fundamental issue in wireless sensor network environments is the efficient location of the required service in the network. The service location protocol is imperative to the design of a wireless sensor network because each network node lacks prior knowledge of the service available in the network [1-7].

Service location in wireless sensor networks is a challenging problem for several reasons. First, due to a lack of infrastructure, there are no well-known servers in a pre-defined network structure. Second, energy scarcity in a network node in a wireless network necessitates the design of new service location protocols that are qualitatively different from those designed for the wired network. Third, in many cases, wireless networks may scale up to thousands of nodes, rendering the location problem even more challenging [8-20].

In pervasive computing, users receive information regarding the environment in real-time; therefore, the sensor network, which is the foundation of pervasive computing, should enable real-time access. In particular, service information is very time critical in pervasive computing; therefore, the service location protocol for the wireless sensor network should provide high accessibility to service information [21-23]. The easiest way to provide high accessibility is to periodically broadcast (flood) service information to the entire network. This method entails major energy consumption, but it is simple and some protocols use this approach.

To reduce the overhead associated with broadcasting, some protocols restrict the flooding area by forwarding packets in a specific direction, as cross shape or restricted regions. These schemes could reduce the broadcasting overhead but still require unnecessary replications if the service information is not popular. Load scalability is the ease with which a distributed system can expand and contract its resource pool to accommodate heavier or lighter loads; it is the ease with which a system or component can be modified, added, or removed to accommodate a changing load. Service location protocols should rapidly provide service information with a large number of users. Therefore, load scalability is an important metric for a service location protocol [24-40].

In this paper, we propose an adaptive square-shaped trajectory (ASST)-based service location method, which is a novel self-configuring, scalable, energy efficient, and robust service location protocol. ASST is based on Geographic Hash Table (GHT) and Trajectory Based Forwarding (TBF). GHT maps the geographic position of a sensor network field to a hash table. In GHT, the sensor node closest to the position where is computed by hash function is responsible for a set of key and data [6,8,9,11]. ASST stores service information in groups of sensor nodes, called a trajectory. A node wishing to publish (advertise) service information obtains a position through the hash function, and it then uses geographic-aided routing such as GPSR (Greedy Perimeter Stateless Routing) to store service information to the trajectory surrounding the hashed position, as in GHT [6,7]. ASST uses TBF to form a trajectory storing the service information. Replication between nodes in the trajectory reduces the network load on a node because queries from users are distributed to several nodes in the trajectory. To further distribute the network load, ASST adjusts the range and size of the trajectory in proportion to the frequency of user queries. In the next section, we review related work. Section 3 describes ASST, and Section 4 provides performance evaluation. We conclude the paper in Section 5.

2. Related Work

Conventional solutions related to this paper can be classified into the following two approaches: Data Storage architecture in a wireless sensor network and Service Location protocols in an *ad hoc* network, as shown in Table 1 [4,5,15,22,34,37,40].

Table 1. Classification of Data Storage Schemes and Service Location Protocols.

	Data Storage Scheme	Service Location Protocol
Wired Networks	External Storage	Service Directory
Wireless Networks	Internal Storage	Flooding
	Data Centric Storage	FMMS
	GHT	GCLP

In the early days of sensor networks, the data sensed by sensor nodes were collected by a base station and stored externally. Here, external storage means that the node providing the storage space is located separately from the sensor networks. For users on a wired network seeking to access and use the sensed data, there are no problems associated with the external storage. However, if users are mobile, it is difficult to access the external storage on the wired network. To address this problem, internal storage architectures are proposed. In internal storage, each sensor node saves the sensed data to its local storage. Users obtain the data by directly querying the sensor network. Internal storage architecture can reduce the data collection overhead; however, users have to query the entire network to find the data. Data-centric storage architecture provides fast data dissemination by storing the data on the basis of its name. Data-centric storage is an enhanced version of data-centric routing. A data-centric routing scheme presented firstly is directed diffusion. Directed diffusion uses flooding to advertise the interests from sinks to sources throughout the network [4,13]. GHT is a type of data-centric storage architecture. GHT is based on the Distributed Hash Table (DHT) that is the results of research efforts on peer-to-peer (P2P) computing networks. GHT was proposed for data-centric storage with geographic information in a sensor network. GHT is a geographic hash table system that hashes keys into geographic points and stores the key-value set at the sensor node closest to the hashed point. GHT uses geographic perimeter routing to identify a packet home node. GHT provides fast access to the data in the sensor nodes but does not take account of availability and scalability. In GHT, only the home node responds to user queries. This causes a concentration of network load and reduces the energy of the home node [6,22].

Ad-hoc networks and DHT share key characteristics in terms of self organization, decentralization, redundancy requirements, and limited infrastructure. However, node mobility and the continually changing physical topology pose a special challenge to scalability and the design of a DHT for mobile ad-hoc network. Using DHT over wireless sensor networks has gained a lot of attention in the research arena recently. In wireless sensor network, the most important issue in routing is to gather the routed information coming from sensor nodes to the sink node regardless of the identity of the donating node. The problem in this context is to locate efficiently the sensor node, which holds the data item with the minimum number of intermediate nodes to save network energy.

A ScatterPastry platform based on Pastry DHT as an overlay routing platform for distributed applications over wireless sensor network using Scatterweb nodes, a real-world wireless sensor platform was proposed in [27]. A topology-based distributed hash tables (T-DHT) as an infrastructure

for data-centric storage, information processing, and routing in ad hoc and sensor networks was introduced in [28]. T-DHTs do not rely on location information and work even in the presence of voids in the network. Using a virtual coordinate system, we construct a distributed hash table which is strongly oriented to the underlying network topology. The mobile hash-table (MHT) [29] addresses this challenge by mapping a data item to a path through the environment. In contrast to existing DHT, MHT does not maintain routing tables and thereby can be used in networks with highly dynamic topologies. Thus, in mobile environments it stores data items with low maintenance overhead on the moving nodes and allows the MHT to scale up to several ten thousands of nodes.

In [30], the appropriateness of using DHT routing paths for service placement in an SBON was evaluated, when aiming to minimize network usage. SBONs are one approach to implementing large-scale stream processing systems. A fundamental consideration in an SBON is that of service placement, which determines the physical location of in-network processing services or operators, in such a way that network resources are used efficiently. Service placement consists of two components: node discovery, which selects a candidate set of nodes on which services might be placed, and node selection, which chooses the particular node to host a service. By viewing the placement problem as the composition of these two processes we can trade-off quality and efficiency between them. For this, two DHT-based algorithms for node discovery, which use either the union or intersection of DHT routing paths in the Stream-based overlay networks (SBON), was considered and compared their performance to other techniques.

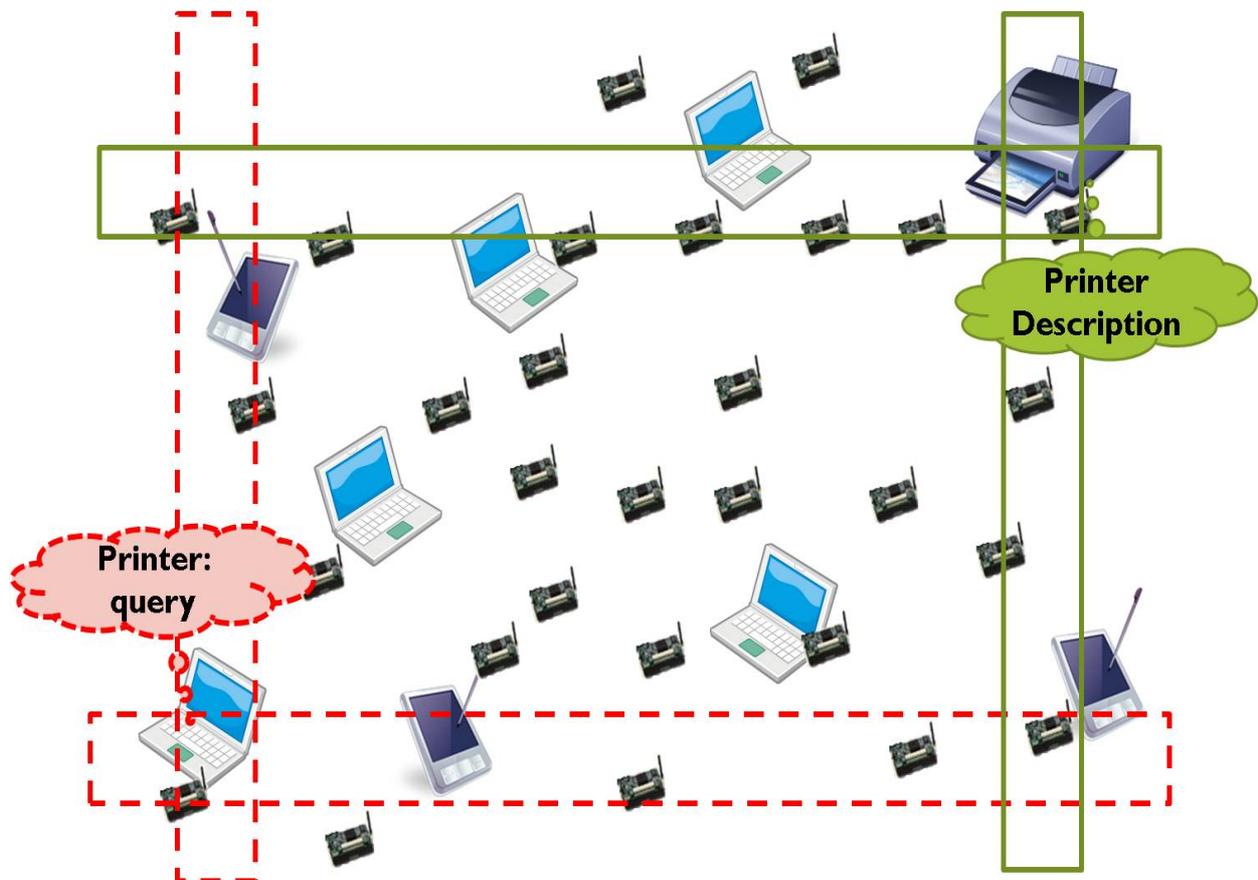
In [31] a GHT based service discovery protocol including the mechanism that constructs topology-aware overlay networks in wireless sensor network was proposed. It does not require a central lookup server and does not rely on multicast or flooding. A Similarity Search Algorithm (SSA) for efficiently processing similarity search queries was proposed in [32]. A data-centric storage structure based on the concept of Hilbert curve and DHT was presented and then an algorithm designed for efficiently probing the most similar data item for the sensor network was proposed. A dynamic geographic hash table for data-centric storage in wireless sensor networks was proposed in [33]. Unbalanced resource utilization problem was addressed by proposing a dynamic GHT solution that relies on two schemes—a temporal-based geographic hash table to achieve overall load balancing among sensor nodes over time and a location selection scheme based on node contribution potential to proactively adapt the system to network dynamics.

An effective hotspot storage management schemes to solve the hotspot storage problem in GHT was proposed in [35]. The scheme included the cover-up and multi-threshold mechanisms. The cover-up mechanism can adjust to another storage node dynamically when a storage node is full, while the multi-threshold mechanism can spread the data into several storage nodes for load balancing of the sensor nodes. Increasing Ray Search (IRS), an energy efficient and scalable search protocol, and k-IRS, an enhanced variant of IRS based on the GHT was proposed in [36]. The priority of IRS is energy efficiency and sacrifices latency whereas k-IRS is configurable in terms of energy-latency trade-off and this flexibility makes it applicable to varied application scenarios. The basic principle of these protocols is to route the search packet along a set of trajectories called rays that maximizes the likelihood of discovering the target information by consuming least amount of energy.

The classical protocols for service location in wired networks rely on a central server called a service directory. The central server advertises its information periodically. Users who want to use a

service connect to the service directory in order to obtain a service description. In wired networks, the network topologies hardly change; therefore, users can access the service directory anytime. However, the central server cannot be used in a wireless network because the network topology in a wireless network frequently changes. The simplest form of service location is global flooding in the network; however, flooding does not scale well. To overcome the weakness of flooding, restricted flooding techniques are developed, such as the Facilitating Match-Making Service (FMMS) and Geography-based Content Location Protocol (GCLP) [5,15]. In FMMS, a service provider advertises the service in a cross-shaped trajectory along the network, as shown in Figure 1.

Figure 1. Service location in FMMS.



Service provider P sends a service advertisement packet in four directions, and the packet is forwarded until it reaches the boundary of the network forming the publish trajectory. A user C who wants to use the service also propagates the query packet in four directions, and the query packet is forwarded until it reaches the boundary of the network packet forming subscribe trajectory similar to advertisement trajectory. The nodes that belong to both trajectories (publish trajectory and subscribe trajectory) reply to user C with the service descriptions. GCLP reduces the query forwarding overhead by stopping the propagation of a query packet when the subscribe trajectory crosses the publish trajectory. FMMS and GCLP reduce the flooding overhead by limiting the flooding to four trajectories; however, both protocols are unable to provide scalability because they do not consider the query amounts, such that the trajectory always has the same size.

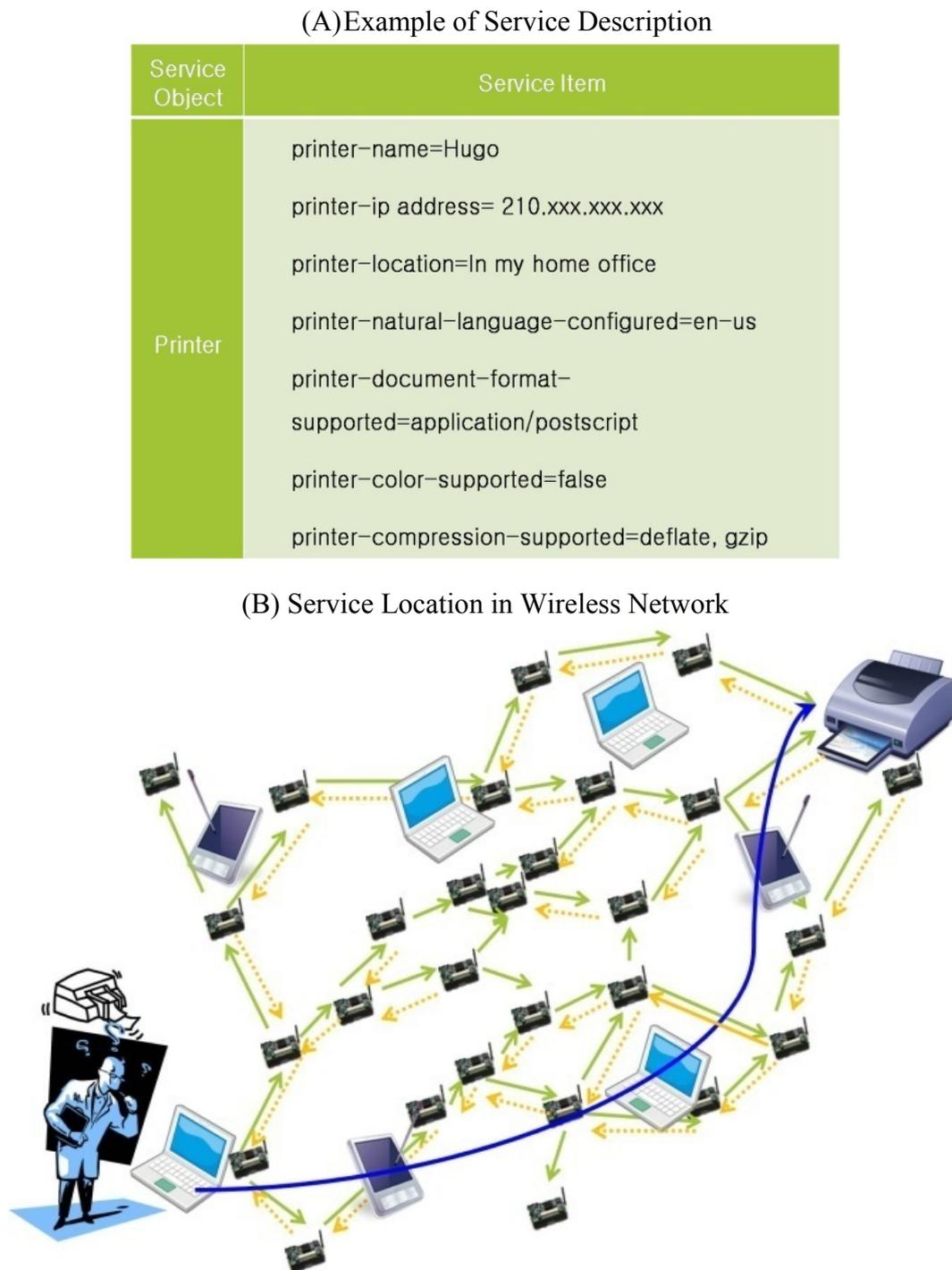
The distance-sensitive service discovery problem in wireless sensor and actor networks was formalized, and a novel localized algorithm, iMesh was proposed in [38]. Unlike existing solutions, iMesh uses no global computation and generates constant per-node storage load. In iMesh, new service providers (*i.e.*, actors) publish their location information in four directions, updating an information mesh such as GCLP. Information propagation for relatively remote services is restricted by a blocking rule, which also updates the mesh structure. Based on an extension rule, nodes along mesh edges may further advertise newly arrived relatively near service by backward distance-limited transmissions, replacing previously closer service location. The final information mesh is a planar structure constituted by the information propagation paths. It stores locations of all the service providers and serves as service directory. Service consumers (*i.e.*, sensors) conduct a lookup process restricted within their home mesh cells to discover nearby services. We analytically study the properties of iMesh including construction cost and distance sensitivity over a static network model.

A service discovery protocol based on hierarchical grid architecture in an ad hoc network which enhanced the GCLP was proposed in [39]. The geographical area was divided into a 2D logical hierarchical grid and the information of available services was registered to a specific location along a predefined trajectory. To enhance resource availability and effective discovery of GCLP, each grid cell selects a directory to cache available services. This work utilizes the transmitting trajectory to improve the efficiency of registration and discovery. First, the service provider registers a service along the proposed register trajectory. The requestor then discovers the service along the discovery trajectory to acquire the service information.

3. ASST

3.1. Basic Concept

This section presents the basic design of ASST that is based on the following assumptions: (1) a vast field is covered by a large number of homogeneous sensor nodes that communicate with each other through short-range radios. (2) Each sensor node is aware of its own location and uses geographic routing such as GPSR to accomplish long distance delivery. (3) Service information means the service description that describes the characteristic of the service, as shown in Figure 2. (4) The storage space on the sensor node is sufficiently large to save the service information. Recently, sensor nodes with a very large memory space, such as RISE (Riverside Sensor) with 1 GB of flash memory, have been developed so that sensor nodes can store a large amount of service information and occupy very little space [17].

Figure 2. Example of service description and service location in a wireless network.

Services have descriptions that include a service name and a service provider ID. Figure 2A shows an example of the service description of a printer. The service description can differ from applications to application and mainly depends on the types of services. Users wishing to use services provided by a wireless sensor network need to obtain the service description. The simple way to obtain the service description is by query flooding the entire network, as shown in Figure 2B. In the figure, a user with a laptop wants to print documents. Query packets with the service name that the user wants to use are broadcasted to all the nodes in the network until they reach the service provider matching the requirement in the query packet. The user connects to the printer server on the basis of the service

description reply sent by the service provider. The flooding is easy and simple, but it may require a long time to discover the required service provider. Furthermore, the flooding could cause a considerable amount of energy consumption on the sensor nodes as a result of broadcast storming. Therefore, efficient service location protocols that provide service information quickly and with low network load are required in a wireless sensor network.

ASST is based on GHT and extends the DHT. Service descriptions that consist of a service name, a service provider ID, *etc.*, are stored in a distributed manner using an algorithm based on DHT. The specific service description is stored in repository nodes that correspond to the hash value of the service name. Geographic routing delivers the information for the service descriptions to and from the repository nodes. When a service provider wants to advertise its service description information on the network, the target point $Q(T_x, T_y)$ hash key corresponding to the service name is first obtained using a well-known hash function such as MD5 and SHA. Once the target point $Q(T_x, T_y)$ hash key is obtained, the service provider finds the repository nodes surrounding the hash value and stores the service description in those nodes. For example, in Figure 3, a node provides a printer service and wants to advertise its service to the network. First, the node finds the repository nodes for the printer by calculating the hash key for the printer service. If the printer's hash key is (11, 28), nodes around (11, 28) become the repository nodes for the printer. A user wanting to use a printer finds the repository nodes for the printer by calculating the hash key of the printer service, same as the service provider.

Figure 3. Example of ASST.

(A) Service Advertisement

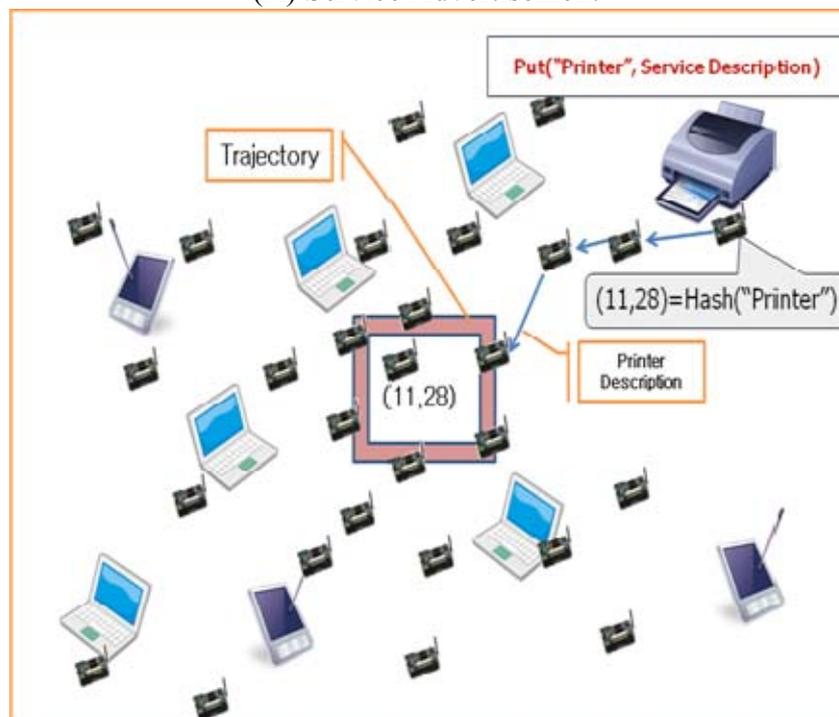
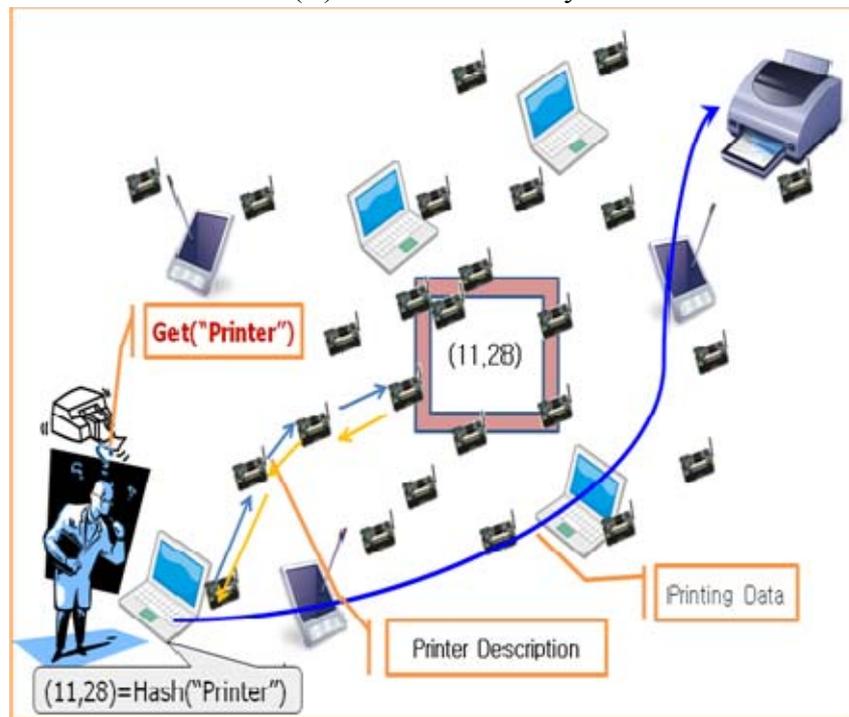


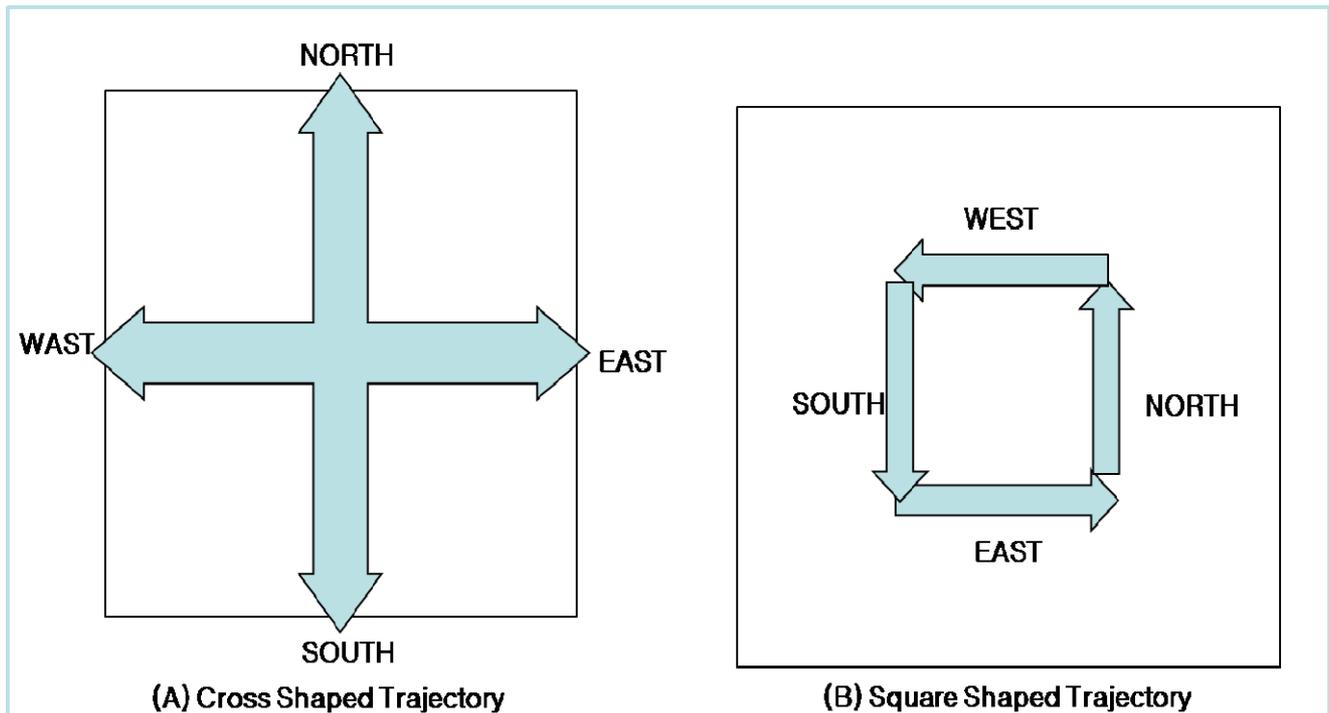
Figure 3. Cont.

(B) Service Discovery



ASST provides a general architecture for service discovery. Both the store and the query operations can be viewed as a general insertion and lookup, and the key can be looked-up in the hash table. Different applications can have different insertion and lookup characteristics. Therefore, different policies should be applied that are based on the characteristics of application. The popularity of data (equivalent to the data query frequency) is one of the most important characteristics of the lookup operation. Different data in an application may have different popularities, and this could lead to an unbalanced load distribution in the existing GHT.

ASST transforms the cross shaped trajectory into the square shaped trajectory to control the number of nodes on the trajectory in proportion of popularity of data as shown in Figure 4. In the cross shaped trajectory, the publish trajectories are forwarded toward the boundary of sensor network field and the shape is fixed regardless of popularity of data. Therefore, it is possible some nodes on the trajectory not to receive any subscription trajectory (query packet) if the popularity of data is very low. Moreover, forwarding the publish/subscription trajectory toward the boundary of sensor network field can be overhead when a sensor network is deployed in large area. On the other hand, the shape of square trajectory can be increased or decreased easily and thus can be said more flexible than cross shaped trajectory. ASST distributes the query processing load to several replicated nodes, this is called trajectory zone. Trajectory zone in ASST is a square-shaped region with replica nodes for the data. A sensor node within the trajectory zone is called a repository node, which is responsible for the data storage and query response. To provide high scalability and accessibility for the data, ASST adjusts the range of the trajectory in proportion to the data's popularity; this is referred to as dynamic trajectory.

Figure 4. Cross Shaped Trajectory and Square Shaped Trajectory.

3.2. Square-Shaped Trajectory

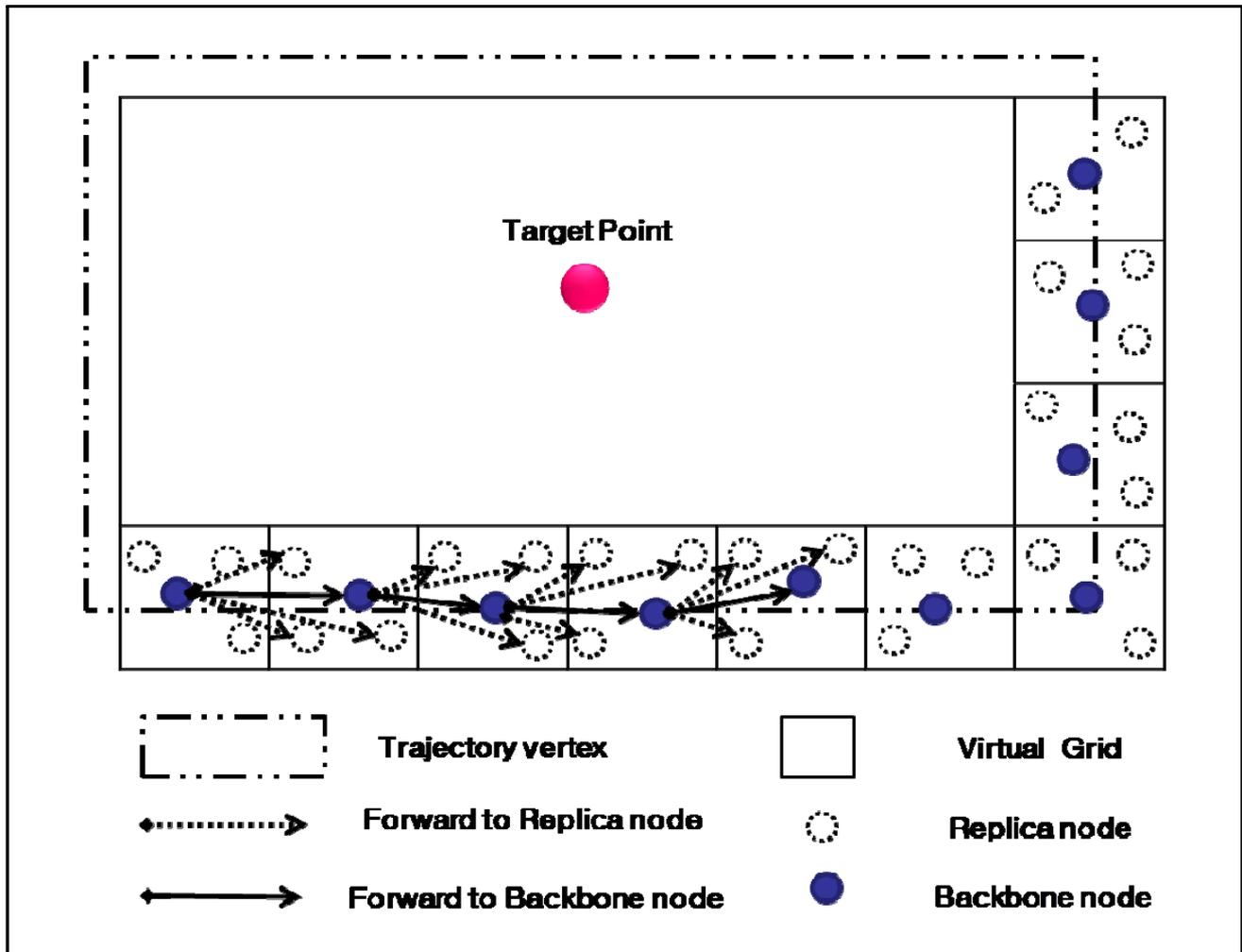
As in the case of many distributed hash table systems, ASST provides a *(key, value)*-based associative memory. Services are named with keys. Both the storage of the service and its retrieval are performed using these keys. Any naming scheme that distinguishes the services that users of the sensor network wish to distinctly identify will suffice in ASST [6]. ASST supports the following two operations:

Put($k; v$) stores v (the observed data) according to the key k , the name of the service.

Get(k) retrieves whatever stored value is associated with key k .

First, the backbone node R_n in the trajectory T_r obtains data from the data packet sent by source S , and it then forwards the data packet to the next backbone node along edge E and toward vertex V in a counterclockwise direction, as shown in Figure 5. By overhearing the data packet forwarded by the backbone node, replica nodes around the backbone node obtain the data and store it. Source node S frequently sends data packets to update the data on the repository node. Each repository node keeps a timer for the data it stores. If the timer expires before any update packet is received from the source, then the data is discarded. To prevent query slippage in the trajectory, we use the method proposed in [9]. The method divides the network into a virtual grid and ensures that the trajectory is constructed through the sequential cells with sensor nodes. The grid line in Figure 5 shows the virtual grid in the network; the trajectory is formed by forwarding the data packet through the sequential cells. In [9], the trajectory was cross shaped; the trajectory in this paper is square shaped. However, the process of trajectory formation is the same for both, only the order differs.

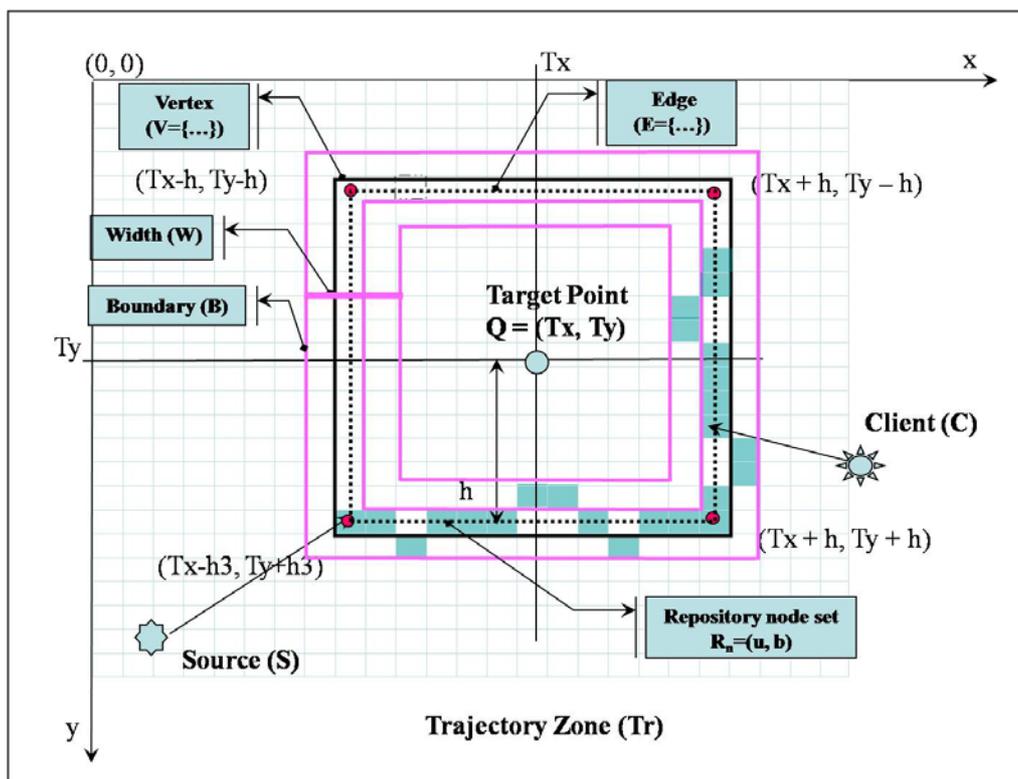
Figure 5. Trajectory formation.



Trajectory Tr is formed away from the target point Q with a distance H (h, h_1, h_2, \dots, h_n) computed in proportion to the data popularity (the same as the query frequency) p by the source node S . In ASST, source nodes only need to know the target point Q and the distance h to form the trajectory Tr . A node that receives a data packet from source S could check whether or not it is involved in trajectory Tr . Trajectory Tr is shaped with vertex V , edge E , and boundary B . Figure 6 shows the elements and formation process of the trajectory Tr . Vertex V of trajectory Tr is a corner of the square with a distance $H(h, h_1, h_2, \dots, h_n)$ from the target point Q (T_x, T_y). The value of vertex V is one of $(T_x - h, T_y - h)$, $(T_x + h, T_y - h)$, $(T_x - h, T_y + h)$, and $(T_x + h, T_y + h)$. Edge E of trajectory Tr is a line connecting two vertexes V . Trajectory Tr has a boundary B at each side of edge E with width W .

Trajectory Tr is a set of data-centric storage nodes, called repository node R_n . Repository node R_n consists of a backbone node b , which lies on the edge E of trajectory Tr , and replica nodes u , which lie between boundary B and edge E . Source S sends a data packet toward the target point. The header of the data packet contains the target point Q (T_x, T_y), range distance h , Source ID, and the Data ID. Each node between the source node and the target point computes the vertex V and edge E for the data packet. If the node is on the edge or vertex of the trajectory's square, then the node becomes the backbone node and starts to form trajectory Tr .

Figure 6. Square-Shaped Trajectory Formation.



3.3. Dynamic Trajectory

To provide scalability for a large sensor network, ASST dynamically adjusts the square range of the trajectory in proportion to the query frequency for the data. If the number of queries for the specific data increases, the range of trajectory Tr also increases. On the other hand, if the number of queries for the data decreases, the range of trajectory Tr decreases. This is known as Dynamic Trajectory.

To decide the distance H which is the range of trajectory Tr , ASST computes the data popularity (query frequency) p for the data. Initially, we gather the number of total queries on all repository nodes R_n . This information can be gathered through the summation of the number of queries at each node and by piggybacking this on the data packet. The repository node feeds back the query information by returning the data packet to the source node. First, the backbone node b in trajectory Tr gathers query counts from the replica nodes around it, and then it piggybacks the total query counts, which is a summation of the query counts, for both itself and the replica nodes. Next, the backbone node that receives the data packet forwarded from the previous backbone node, adds both its query counts and the replica node's query counts to the query counts from the data packet, and then it forwards this to the next backbone node. When the data packet returns to the first backbone node, the data packet is returned to source S , which computes a new distance H .

Figure 7 shows the Dynamic Trajectory in ASST. Dynamic Trajectory starts with a minimum distance h_1 . The source node S sends a data packet with a distance h_1 . At the next update, the source S sends a data packet with a distance h_1 and computes a new distance with the query frequency. The new distance is not applied immediately but at the time of the next update. If the query frequency is increased over the threshold, the distance h is also increased to h_2 . At distance h_2 , if the query

frequency is decreased, the distance is decreased to h_1 ; if the query frequency is increased again, the distance becomes h_3 . When the distance is increased or decreased, a new trajectory is generated with the new repository nodes. We do not need to manually remove the old trajectory because each repository node has an update timer. If the data is not updated until the timer is expired, the node discards the data and this leads to deformation of the trajectory. Table 2 shows the procedures of dynamic trajectory in ASST.

Figure 7. Dynamic Trajectory Development.

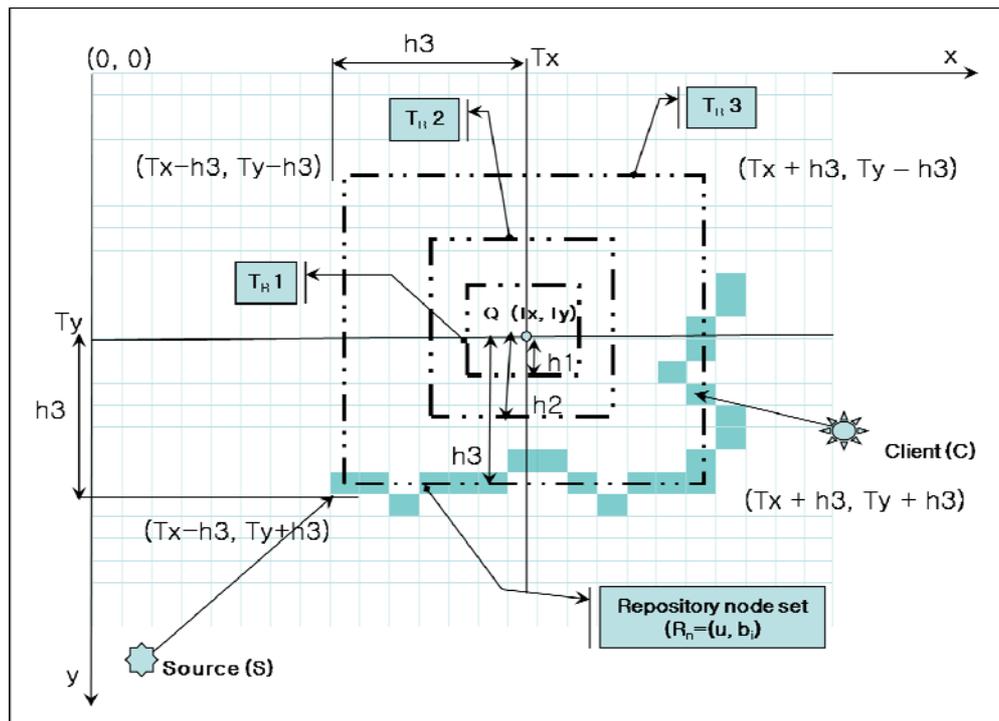


Table 2. Dynamic ASST procedure.

```

procedure put( $k, v$ )
 $TP(Tx, Ty) \leftarrow Hash(k)$ 
 $QF \leftarrow get\_Query\_Freq(k)$ 
 $h \leftarrow QF \times QT^2$ 
 $Vertex[4] \leftarrow \{(Tx - h, Ty - h), (Tx - h, Ty + h), (Tx + h, Ty + h), (Tx + h, Ty - h)\}$ 
 $Min\_Dis \leftarrow MAX\_DIS$ 
for  $i = 1$  to 4 do
   $dis \leftarrow distance(my\_P, Vertex[i])$ 
  if  $dis < Min\_Dis$  then
     $Min\_dis \leftarrow dis$ 
     $Targ et\_Vertex \leftarrow Vertex[i]$ 
     $Direction \leftarrow i$ 
  end if
end for
 $Send(US, k, v, TP, h, Vertex, Targ et\_Vertex, Direction, ET)$ 
return

```

Table 2. Cont.

<pre> <i>procedure receiveVertex(k, v, TP, h, Vertex, T arg et _ Vertex, Direction, ET)</i> <i>if Us = my _ US then</i> <i>return</i> <i>my _ US ← US</i> <i>my _ k ← k</i> <i>my _ v ← v</i> <i>my _ ET ← ET</i> <i>set _ Timer(my _ ET)</i> <i>if Direction = EAST then</i> <i>Direction ← NORTH</i> <i>else if Direction = NORTH then</i> <i>Direction ← WEST</i> <i>else if Direction = WEST then</i> <i>Direction ← SOUTH</i> <i>else if Direction = SOUTH then</i> <i>Direction ← EAST</i> <i>end if</i> <i>T arg et _ Vertex ← Vetext[Direction]</i> <i>GN ← Grid _ Number(Direction)</i> <i>Neighbor[] ← Neighbor _ Set[GN]</i> <i>Backbone ← min dis(TP, h, Neighbor)</i> <i>Send(US, k, v, TP, h, Vertex, T arg et _ Vertex, Direction, Backbone, GN)</i> <i>return</i> </pre>
<pre> <i>procedure receiveRe pository(k, v, TP, h, Vertex, T arg et _ Vertex, Direction, ET, Backbone, GN)</i> <i>if US = my _ US then</i> <i>return</i> <i>end if</i> <i>if my _ GN = GN then</i> <i>my _ US ← US</i> <i>my _ k ← k</i> <i>my _ v ← v</i> <i>myET ← ET</i> <i>set _ Timer(ET)</i> <i>end if</i> <i>if Backbone = my _ ID then</i> <i>T arg et _ Vertex ← Vertex[Direction]</i> <i>Neighbor[] ← Neighbor _ Set[Direction]</i> <i>Backbone ← min dis(TP, h, Neighbor)</i> </pre>

Table 2. Cont.

<pre> end if Send(US,k,v,TP,h,Vertex,Target_Verx,Direction,Backbone) return </pre>
<p> k : key of value v : value TP : Target Point QF : Query Frequency my_P : position of source US : Update Sequence ET : Expire Time GN : Next Grid Number </p>

3.4. Analysis

3.4.1. Robustness

Sensor networks consist of small fragile sensor nodes with limited resources such as computing power, energy, and network bandwidth. Sensor nodes can easily fail as a result of a physical attack or through energy dissipation. Therefore, sensor nodes should be protected, and systems running on sensor nodes should provide failure tolerance. The service location system should also be able to rapidly provide service information to the user, even if there are node failures in the network. ASST provides failure tolerance with a virtual grid when ASST forms a trajectory and the sensor node forwards the query packet, as shown in Figure 8.

Figure 8. Failure Avoidance and Recovery.

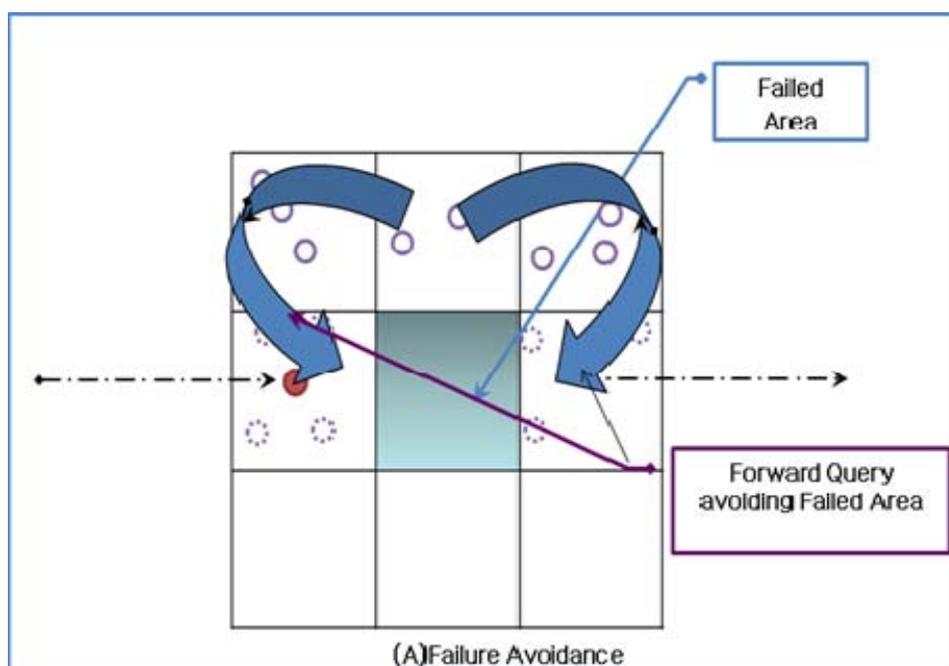
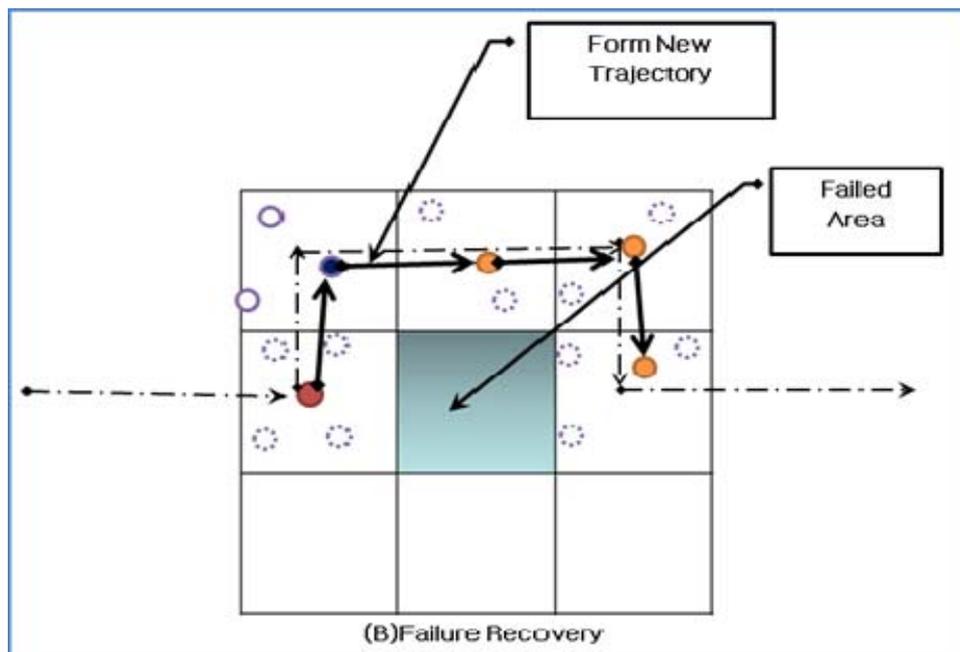


Figure 8. Cont.



A virtual grid on the trajectory consists of a backbone node and several replica nodes; other nodes in the virtual grid can respond to a query in the event of node failure, as long as at least one node is alive in the grid. If all the nodes in the grid fail, the query is forwarded to other grids by geographic routing, thus avoiding the failed area. Both the trajectory formation and query forwarding are based on the virtual grid; therefore, the query cannot miss the trajectory, as shown in [9]. When the service provider updates the trajectory, the trajectory is recovered by making a detour around the failed area. This technique can be extended for multiple grid failures.

3.4.2. Load scalability

To be load scalable, a scheme has to offer uniform processing times irrespective of the loads. The service location protocol should be scalable to accommodate the variable popularity of a service. ASST increases the distance H of the trajectory in proportion to the query frequency in order to provide a uniform query processing time to the user. The number of virtual blocks on the trajectory with a distance H is $NUM_{VB} = \frac{4(2H)}{W}$, and the number of queries that one virtual block can process

during T is $NUM_{proc} = \frac{T \times SN_{Avg}}{t_{proc}}$, where t_{proc} is the processing time of one query on the sensor node

and SN_{Avg} is the average number of sensor nodes in a virtual block. When the total number of queries during T is NUM_{query} , the number of virtual block that requires to process the NUM_{query} is

$NUM_{reqVB} = \frac{NUM_{query}}{NUM_{proc}}$ and thus the distance H_{query} is $H_{query} = \frac{W \times NUM_{query}}{8 \times NUM_{proc}}$. The required

processing time for NUM_{query} is $t_{req} = NUM_{query} \times t_{proc}$. The query processing time per node is:

$$\begin{aligned}
 t_{total} &= \frac{t_{req}}{SN_{total}} = \frac{NUM_{query} \times t_{proc}}{SN_{Avg} \times NUM_{VB}} = \frac{NUM_{query} \times t_{proc}}{SN_{Avg} \times \frac{8H_{query}}{W}} = \frac{NUM_{query} \times t_{proc}}{SN_{Avg} \times \frac{8 \times W \times NUM_{query}}{8 \times NUM_{proc}}} \\
 &= \frac{NUM_{query} \times t_{proc}}{SN_{Avg} \times \frac{NUM_{query}}{NUM_{proc}}} = \frac{NUM_{query} \times t_{proc}}{SN_{Avg} \times \frac{NUM_{query}}{T \times SN_{Avg}}} = T \\
 &\qquad\qquad\qquad t_{proc}
 \end{aligned}$$

Therefore, there is no delay time on the repository node.

3.4.3. Time and message

For the data transmissions from a child sensor node to a parent node, CSMA is used to reserve the channel. We assume that each node knows the number of contending neighbor nodes (m) and contends the channel with the optimal probability $p = 1/m$. The probability that one contending node wins the channel is $p_{succ} = (1 - 1/m)^{m-1}$. Since the number of slots needed until the successful reservation is a geometric random variable, the average number of contending slots (ACS) is given by:

$$ACS = \frac{1}{(1 - \frac{1}{M})^{M-1}}$$

where M is the Average number of Neighbor nodes.

One node on the grid included to the Trajectory Zone has to forward the packet to next virtual block and therefore, the required time slots for Trajectory Zone (STZ) is given by:

$$STZ = ACS \times NUM_{VB} = \frac{1}{(1 - \frac{1}{M})^{M-1}} \times \frac{8H}{W}$$

The main part of procedure in Table 2 is finding the next node which is closest to the trajectory Edge and the time for this routine is in proportion to the number of nodes in a virtual zone, which is in proportion to the width of Trajectory Zone (W). Sorting with distance and finding a node minimum distance requires $O(n \log n)$ time. The required time for inter communication between backbone nodes is in proportion to the width of Trajectory Zone (W) and the distance H as shown above and thus can be also $O(n \log n)$. As a consequence, the total time complexity is $O(n \log n)$.

The number of message for Trajectory Zone formation is in proportion to the number of virtual block in Trajectory Zone since every backbone node in virtual blocks has to forward the formation message to next virtual block and thus the number of message for Trajectory Zone formation is given by $O(N)$.

4. Performance Evaluation

In Section 3, we proposed ASST, a new mechanism for data dissemination based on DHT and TBF. In this section, we evaluate the performance of our proposed mechanism in ns-2 simulations. Ns-2 supports detailed simulation of mobile, wireless networks. Our simulation uses an 802.11 radio with a 30 m radio range, rather than the 250 m radio range of IEEE-compliant hardware; this choice is similar

to that made in the evaluation of GHT with a 40 m radio range. ASST was implemented on the basis of GPSR. 900 sensor nodes were deployed in a 600 m \times 600 m field in grid topology. The distance between sensor nodes was 20 m so that each sensor node has 8 neighboring nodes. The sensor nodes at the border line of the field were set as query nodes, frequently sending queries. The target point was set at the center of the field. Each sensor node consumes 0.5 w of power when it sends and 0.2 w of power when it receives. Table 3 provides details of the simulation parameters. RXThresh is a receive threshold, CPThresh capture threshold, and CStresh carrier-sensing Threshold.

Table 3. Simulation parameters.

Parameter		Value	Parameter		Value
Radio Propagation		Shadowing	Routing Protocol		GPSR
MAC		802.11	Energy Consumption	Idle Power	0.1 w
Queue		DropTail		Rx Power	0.2 w
Antenna		OmniAntenna		Tx Power	0.5 w
Network size		600 m \times 600 m	Nodes		900 nodes
Radio Parameter	CPThresh	10.0	Packet size		60 bytes
	CStresh	1.559e-11	Antenna Parameter	X	0
	RXThresh	3.652e-10		Y	0
	Pt	2.5872e-4		Z	1.5

Scalability, as a property of systems, is generally difficult to define; however, it is essential to define the specific requirements for scalability based on the dimensions that are deemed important. In telecommunications and software engineering, scalability is a desirable property of a system, a network, or a process; it indicates the ease with which the system can either handle growing amounts of work, or can be readily enlarged. For example, it can refer to the capability of a system to increase total throughput under an increased load. An algorithm, design, networking protocol, program, or other system is said to be scalable if it is suitably efficient and practical when applied to large situations (e.g., a large input data set or a large number of participating nodes in the case of a distributed system). If the design fails when the quantity increases then it does not scale. In particular, load scalability means the ability of a distributed system to expand easily and to contract its resource pool to accommodate heavier or lighter loads. Alternatively, it is the ease with which a system or component can be modified, added, or removed to accommodate a changing load. In this paper, we use load scalability as an evaluation metric.

We evaluate the performance of ASST with a diverse query frequency. We compare ASST with GHT and GCLP. To evaluate the scalability of ASST, we varied the query frequency from 0.5 queries per second to 100 queries per second. To reflect query processing, each query has a query processing time of 100 ms. A replica node receiving the query packet has to wait for 100 ms before responding. To provide load scalability, each query has to maintain the response time of 100 ms. The query was uniformly distributed among sensor nodes.

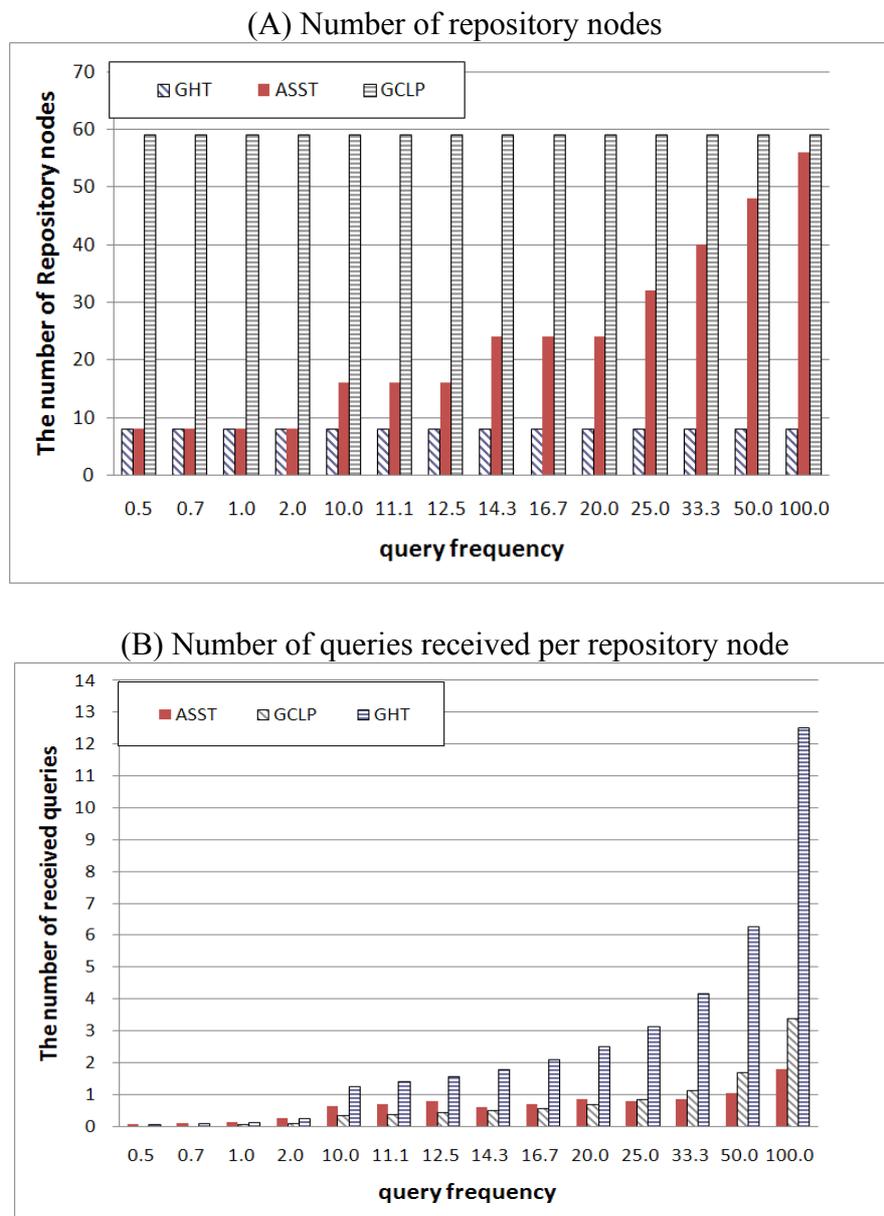
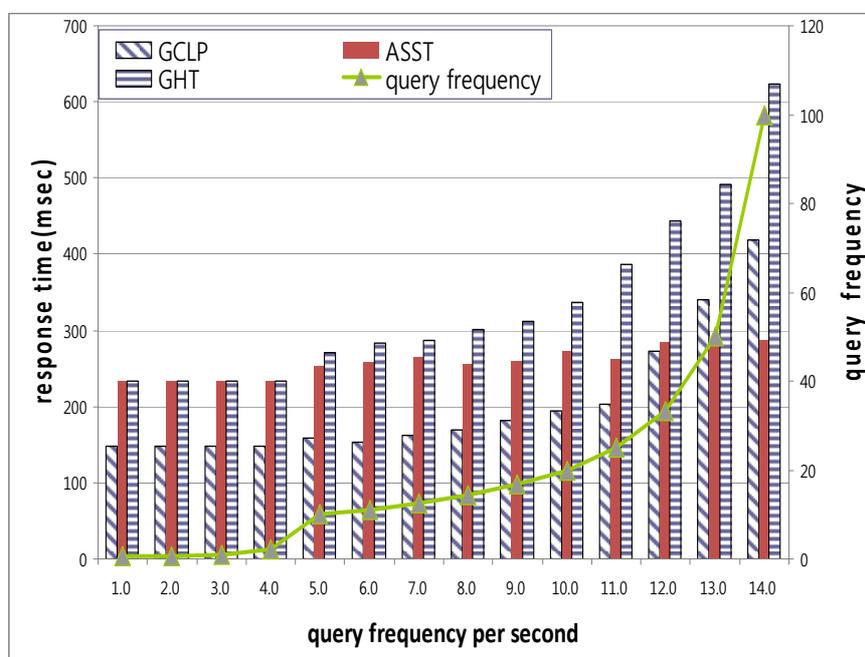
Figure 9. Number of repository nodes and the number of queries received per repository node.

Figure 9 shows the number of repository nodes and the number of queries received per repository node given several query frequencies. GHT saves information into very few repository nodes—home nodes and neighboring nodes surrounding the target point—and GCLP uses many repository nodes forming a trajectory from end-to-end in the network. Neither GHT nor GCLP take account of the query frequency; therefore, the number of repository nodes is fixed in spite of the increased query frequency. This leads to an increase in the number of queries received per node. On the other hand, ASST varies the size of the trajectory in proportion to the query frequency, so that the number of repository nodes in ASST is increased in proportion to the increased query frequency. Therefore, the number of received queries per node in ASST is smaller than that in GHT and GCLP. GHT can be regarded as minimum replication, and GCLP can be regarded as maximum replication. It should be noted that the number of repository nodes in GCLP and ASST is similar when the query frequency is high, but the number of received queries per node in GCLP is higher than that in ASST due to the two

matching points in GCLP. In GCLP, both the publish trajectory and the subscribe trajectory form a cross, making two matching points. Therefore, the number of received queries is higher than that in ASST.

The number of queries per node can affect the response time. Figure 10 shows the response time with an increased query frequency. When the query frequency is low, GCLP shows a shorter response time than ASST or GHT because GCLP has more repository nodes than GHT or ASST. Moreover, GCLP forms a trajectory crossing the network so that GCLP occupies a larger area than GHT or ASST, and this leads to a reduction in the hop counts between the repository and query nodes. However, as the query frequency increases, the response time also increases. When the query frequency is very high, the response time is longer than that in ASST. This is caused by the larger number of received queries. However, GHT and ASST have the same number of repository nodes and the same hop counts between the repository and query nodes when the query frequency is low, so that the response times of both is the same. However, when the query frequency is increased, the response time of GHT is increased drastically while the response time of ASST is almost uniform. This is due to the difference between GHT and ASST in terms of the number of received queries per node. GHT always has a fixed number of repository nodes so that the number of received queries per node increases, and this leads to an increased response time. On the other hand, ASST increases the number of repository nodes in proportion to the query frequency so that the number of received queries per node and response time are almost uniform.

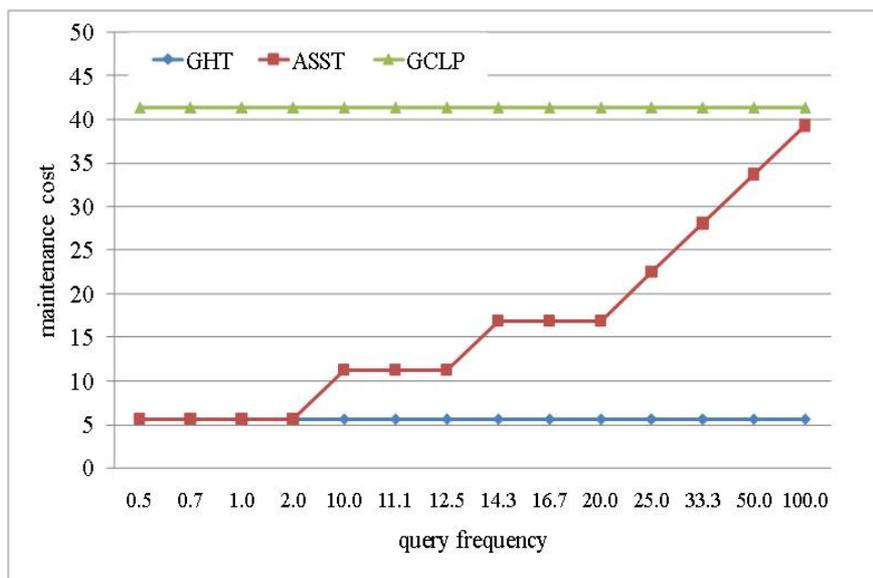
Figure 10. Response time with increased query frequency.



Replicating data in several sensor nodes requires energy costs for packet transmission. More nodes participate in replication, and more energy is consumed because energy consumption is increased whenever a node transmits a packet. Figure 11 shows the energy consumption for repository construction. As mentioned earlier, the number of repository nodes in GHT and GCLP is fixed in spite

of query frequency change. GHT maintains a minimum number of repository nodes around the target point so that the energy consumption is lower than that with GCLP or ASST. In contrast, GCLP maintains repository nodes between the network boundaries in a cross shape so that energy consumption is higher than that under GHT or ASST. On the other hand, ASST varies the number of repository nodes in proportion to the query frequency so that the energy consumption is low when the query frequency is low, and the energy consumption is increased in proportion to the query frequency. The energy consumption is similar with GHT when the query frequency is low, and the energy consumption is increased towards the GCLP as the query frequency increases.

Figure 11. Energy consumption for storage construction.



5. Conclusions

In this paper, we have proposed a new energy efficient and scalable data dissemination method, *i.e.*, ASST, based on DHT and TBF. ASST is a type of data-centric storage system for sensor networks. ASST provides fast access to the data stored in a sensor network by using the distributed hash function. In ASST, the source node stores data at the nodes forming a trajectory around the target position computed by the hash function. A client sends a query packet to the position computed by the hash function that is the same as the source node. By storing data at several repository nodes, ASST distributes the network load caused by query packets.

ASST also provides scalability for a large sensor network. When the query frequency is increased, the range of the trajectory is also increased to reduce the network load at the repository node. By adjusting the range of the trajectory in proportion to the query frequency, ASST can ensure a fast response time for multiple clients and reduce the network load assigned to a sensor node. Such capabilities result in an increased life span for the sensor network.

The main focus of ASST is providing load scalability with constant response time by adjusting the width of trajectory in proportion to the query frequency under uniform distribution of query, however the query distribution might not be uniform over network and thus increasing or decreasing four edges with same distance H can cause load unbalance under non-uniform distribution of query. In the future

work, we will consider the adjustment of distance H of four edges in trajectory zone in separated manner under with under non-uniform distribution of query.

References

1. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. A Survey on Sensor Networks. *IEEE Commun. Mag.* **2002**, 102-114.
2. Henn, H.; Hepper, S.; Rindtorff, K.; Schack, T.; Burkhardt, J. *Pervasive Computing: Technology and Architecture of Mobile Internet Applications*, 1st ed.; Addison-Wesley Professional: Boston, MA, USA, 2002.
3. Niculescu, D. Communication Paradigms for Sensor Networks. *IEEE Commun. Mag.* **2005**, 43, 116-122.
4. Seada, K.; Helmy, A. Rendezvous Regions: A Scalable Architecture for Service Location and Data-Centric Storage in Large-Scale Wireless Networks. In *proceedings of 18th International Parallel and Distributed Processing Symposium (IPDPS'04)*, Santa Fe, New Mexico, April 26–30, 2004.
5. Tchakarov, J.B.; Vaidya, N.H. Efficient Content Location in Wireless Ad Hoc Networks. *IEEE International Conference on Mobile Data Management*, Berkeley, CA, USA, January 19–22, 2004; p. 74.
6. Ratnasamy, S.; Karp, B.; Yin, L.; Yu, F.; Estrin, D.; Govindan, R.; Shenker, S. GHT: A Geographic Hash-table for Data centric Storage in Sensornets. In *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, Atlanta, GA, USA, September 28, 2002; pp.78-87.
7. Karp, B.; Kung, H.T. *GPSR: Greedy Perimeter Stateless Routing for Wireless Networks*; A870044; MobiCom: Houston, TX, USA, 2000.
8. Capone, A.; Pizziniaco, L.; Filippini, I. A SiFT: An Efficient Method for Trajectory Based Forwarding. In *Proceedings of Wireless Communication Systems 2nd International Symposium*, Siena, Italy, September 5–7, 2005; pp. 135-139.
9. Tscha, Y.; Caglayan, M.U. Query Slipping Prevention for Trajectory-based Publishing and Subscribing in Wireless Sensor Networks. *Comp. Commun.* **2006**, 29, 1979-1991.
10. Machado, M.; Goussevskaia, O.; Mini, R.A.F.; Rezende, C.G.; Loureiro, A.A.F.; Mateus, G.R.; Nogueira, J.M.S. Data Dissemination in Autonomic Wireless Sensor Networks. *IEEE J. Sel. Area. Commun.* **2005**, 23, 2305-2319.
11. Akkaya, K.; Younis, M. A survey on routing protocols for wireless sensor networks. *Ad Hoc Netw.* **2005**, 3, 325-349.
12. Liu, X.; Huang, Q.; Zhang, Y. *Combs, Needles, Haystacks: Balancing Push and Pull for Discovery in Large-scale Sensor Networks*; ACM Press: New York, NY, USA, 2004; pp. 122-133.
13. Intanagonwiwat, C.; Govindan R.; Estrin, D. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks, submitted to ACM MobiCom 2000.
14. GridwiseTech. Principles of Scalable Systems—Describing Technologies/layers that Are Components of Scalable System. Available online: http://www.gridwisetech.com/scalability_expert (accessed November 2007).

15. Aydin, I.; Shen, C.C. Facilitating Match-making Service in Ad Hoc and Sensor Networks Using Pseudo Quorum. In *Proceedings of the 11th IEEE International Conference on Computer Communications and Networks (ICCCN)*, Miami, FL, USA, October 14–16, 2002.
16. Sailhan, F.; Issarny, V. Scalable service discovery for MANET. In *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications*, Kauai Island, HI, USA, March 8–12, 2005.
17. Mitra, A.; Banerjee, A.; Najjar, W.; Zeinalipour-Yazti, D.; Kalogeraki, V.; Gunopulos, D. High-Performance Low Power Sensor Platforms Featuring Gigabyte Scale Storage. In *Proceedings of IEEE/ACM 3rd International Workshop on Measurement, Modelling, and Performance Analysis of Wireless Sensor Networks*, San Diego, CA, USA, July 21, 2005.
18. Jurczyk, P.; Xiong, L.; Sunderam, V. DObjects: Enabling Distributed Data Services for Metacomputing Platforms. *ICCS*. Krakow, Poland, June 23–25, 2008; pp.136–145.
19. Yuksel, M.; Pradhan, R.; Kalyanaraman, S. An Implementation Framework for Trajectory-based Forwarding in Ad-Hoc Networks. In *Proceedings of IEEE ICC*, June 20–24, 2004; pp. 4062–4066.
20. Park, S.; Lee, D.; Lee, E.; Yu, F.; Choi, Y.; Kim, S-H. A Communication Architecture to Reflect User Mobility Issue in Wireless Sensor Fields. In *Proceedings of IEEE Communications Society Subject Matter Experts for publication in the WCNC*, Kowloon, HongKong, March 11–15, 2007; pp.3376–3381.
21. Krishnamachari, B.; Ahn, J. Optimizing Data Replication for Expanding Ring-based Queries in Wireless Sensor Networks. *USC Computer Engineering Technical Report*; CENG-05-14, 2005.
22. Shenker, S.; Ratnasamy, S.; Karp, B; Govindan, R.; Estrin, D. Data-Centric Storage in Sensornets. *ACM SIGCOMM, Comp. Commun. Rev.* **2003**, *33*, 137–142.
23. Ball, R. Content Sharing for Mobile Devices. *CoRR* **2008**, abs/0809.4395.
24. Gjermundrød, H.; Bakken, D.E.; Hauser, C.H.; Bose, A. GridStat: A Flexible QoS-Managed Data Dissemination Framework for the Power Grid. *IEEE Trans. Power Delivery* **2009**, *24*, 136–143.
25. Liang, S.H.L. A New Fully Decentralized Scalable Peer-to-Peer GIS Architecture. In *Proceedings of ISPRS Congress*, Beijing, China, July 3–11, 2008; p. 687.
26. Bondi, A.B. Characteristics of scalability and their impact on performance. In *Proceedings of the 2nd International Workshop on Software and Performance*, Ottawa, Canada, September 17–20, 2000; pp. 195–203.
27. Al-Mamou, A.; Labiod Houda. ScatterPastry: An Overlay Routing Using a DHT over Wireless Sensor Networks. In *Proceedings of The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007)*, Jeju Island, Korea, October 11–13, 2007; pp.274–279.
28. Landsiedel O.; Lehmann K.; Wehrle K. T-DHT: Topology-based Distributed Hash Tables. In *Proceedings of the Fifth IEEE International Conference on Peer-to-Peer Computing (P2P'05)*, Konstanz, Germany, August 31–September 2, 2005; pp. 143–144.
29. Landsiedel, O.; Gotz, S.; Wehrle, K. Towards Scalable Mobility in Distributed Hash Tables. In *Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing*, Cambridge, UK, October 2–4, 2006; pp. 203–209.
30. Pietzuch, P.; Shneidman, J.; Ledlie, J.; Welsh, M.; Seltzer, M. Roussopoulos, M. Evaluating DHT-Based Service Placement for Stream-Based Overlays. In *Proceedings of the 4th International Workshop on Peer-to-Peer Systems (IPTPS'05)*, Ithaca, NY, USA, February 24–25, 2005.

31. Jung, J.; Lee, S.; Kim, N.; Yoon, H. Efficient Service Discovery Mechanism for Wireless Sensor Networks. *Comput. Commun.* **2008**, *31*, 3292-3298.
32. Su, I.; Chung, Y.; Lee, C. Finding Similar Answers in Data-Centric Sensor Networks. *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (Sutc 2008)*, Taichung, Taiwan, June 11–13, 2008. pp. 217-224.
33. Le, T.; Yu, W.; Bai, X.; Xuan, D. A Dynamic Geographic Hash Table for Data Centric Storage in Sensor Networks. *In Proceeding of IEEE Wireless Communications and Networking Conference (WCNC 2006)*, Las Vegas, NV, USA, April 3–6, 2006.
34. Liao W.; Wu W.; A cover-up scheme for data-centric storage in wireless sensor networks, *In Proceeding of IEEE Symposium on Computers and Communications (ISCC 2007)*, Aveiro, Portugal, July 1-4. 2007.
35. Liao W.; Wu W. Effective Hotspot Storage Management Schemes in Wireless Sensor Networks. *Comput. Commun.* **2008**, *31*, 2131-2141.
36. Rachuri, K.; Murthy, S. Energy Efficient and Scalable Search in Dense Wireless Sensor Networks. *IEEE Trans. Comput.* **2009**, *58*, 812-826.
37. Satoshi, M.; Kazutoshi, F.; Hideki, S. Implementation Methodology of Geographical Overlay Network Suitable for Ubiquitous Sensing Environment. *J. Inform. Process.* **2008**, *16*, 80-92.
38. Li, X.; Santoro, N.; Stojmenovic, I. Localized Distance-Sensitive Service Discovery in Wireless Sensor and Actor Networks. *IEEE Trans. Comput.* **2009**, *58*, 1275-1288.
39. Tsai, H.; Chen, T.; Chu, C. Service Discovery in Mobile Ad Hoc Networks Based on Grid. *IEEE Trans. Veh. Technol.* **2009**, *58*, 1528-1545.
40. Mann, C.; Baldwin, R.; Kharoufeh, J.; Mullins, B. A Trajectory-Based Selective Broadcast Query Protocol for Large-Scale, High-Density Wireless Sensor Networks. *Telecommun. Syst.: Model. Anal. Des. Manag.* **2007**, *35*, 67-86.

© 2010 by the authors; licensee MDPI, Basel, Switzerland. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).