# Secure and energy efficient geocast protocol for sensor networks with misbehaving nodes

Young-Chul Shim

***Abstract*—**Geocast sends packets to all sensor nodes within a specified geographical region in order to gather data from sensor nodes in that region and is an important mechanism in sensor networks. In this paper we first introduce an energy-efficient geocast protocol. The proposed protocol builds a multicast tree connecting geocast nodes using an energy efficient broadcasting technique without making any restrictions on the shape of the geocast region. The proposed protocol reduces the energy consumption during the phase of sending commands to the sensor nodes in a geocast region. It also facilitates in-network data aggregation and, therefore, helps save energy during the data reporting phase. Then we modify the proposed protocol to include security mechanisms to protect the multicast tree and data being transferred over this tree not only from outside attackers but also compromised insider attackers.

***Keywords*—**Energy-Efficiency, Geocast, Misbehaving nodes, Security, Sensor Networks.

## I. Introduction

SENSOR networks have been used for a wide range of applications including environment monitoring, traffic surveillance, military sensing, and information gathering. Their main purposes are to monitor an area, including detecting, identifying, localizing, and tracking one or multiple objects of interest. A sensor network consists of one of multiple data center called a sink node and many low-cost and low-powered sensor devices, called sensor nodes. Each sensor node has the ability of sensing data, processing data, and communicating with others via radio transceivers. The sink node, equipped with a database system, sends queries or control commands to sensor nodes and collects information from sensors. The communication between the sink and sensor nodes relies on the relay by intermediate sensor nodes [1].

Because sensor nodes are microelectronic devices, they can only be equipped with limited power source. Therefore, energy conservation becomes one of the most important issues when developing routing protocols for sensor networks [2], [3]. Techniques such as in-network data aggregation are needed to reduce energy consumption in sensor nodes.

In sensor networks a group of sensor nodes in a certain geographic region may cooperate to monitor an object within that region. So multicasting to all the sensor nodes in that region becomes an essential mechanism. Geocast, a variant of conventional multicast, sends packets to all the nodes within a specified geographical region [4]. To determine the geocast group membership, each node is required to know its own physical location, i.e., its geographic coordinates, which may be obtained using a system such as the Global Positioning System (GPS) [5].

In this paper, we first introduce a new geocast protocol in sensor networks. We do not make any restrictions on the shape of the geocast region. The proposed protocol reduces energy consumption during the phase of sending commands from the sink node to the sensor nodes in a geocast region and also facilitates in-network data aggregation and, therefore, helps save energy during the phase of reporting sensor data to the sink node.

Security attacks can come from either illegal outside nodes or legal inside nodes which have been captured and compromised by enemies. The latter nodes are called misbehaving nodes or malicious nodes and attacks from them are more difficult to detect than those from outside attackers. After a node A sends a message to a node B, A can monitor B’s behavior. If B makes illegal behaviors such as forwarding an illegally modified message or refusing to forward the message, A can detect this and notify B’s anomalous behavior to other nodes. But upon listening to this report, B will send a refutation message. Receiving these two conflicting messages, other nodes will not be able to judge which node is the misbehaving node. To resolve this situation, we introduce the concept of watch nodes. A watch node for nodes A and B listens to and stores message from A and B. When a dispute between A and B occurs, the watch node helps decide which node is the malicious node by analyzing stored information.

In this paper we extend the basic geocast protocol so that it can protect commands and data from both outside and inside attackers. In the extended protocol, a sender monitors the message forwarding of the receiver to detect the receiver’s malicious behavior and a watch node monitors both the sender and the receiver to resolve a possible conflict occurring between them.

The rest of the paper is organized as follows. Section 2 surveys the related works. Section 3 describes basic geocast protocol without security mechanism. Section 4 explains the watch node concept. Section 5 explains how the basic geocast

Manuscript received March 20, 2009. This work was supported by the grant No. R01-2006-10073-0 from the Basic Research Program of the Korea Science & Engineering Foundation and the research grant from Hongik University.

Young-Chul Shim is with Hongik University, Seoul, Republic of Korea (phone: +82-2-320-1695; fax: +82-2-332-1653: e-mail: shim@cs.hongik.ac.kr).

protocol is extended to provide security services and is followed by conclusions in Section 6.

## II. Related Works

In this section we first survey related works on geocast and efficient broadcasting techniques for mobile ad hoc networks and then we describe security issues in geocast.

### *A. Geocast*

In general, geocast protocols consist of two phases. In the first phase a packet is delivered from the source to one or more nodes in the geocast region. Then the packet is broadcast to all the nodes in the geocast region. Although a geocast protocol consists of two phases, most of proposed geocast protocols for Mobile Ad-hoc NETworks(MANETs) focus on the protocol for the first phase and assume the use of flooding for the second phase. Yao et al classified geocast protocols into three categories: flooding-based protocols, routing-based protocols, and cluster-based protocols [5].

Flooding-based protocols use flooding or a variant of flooding to forward geocast packets from the sink to the geocast region [5]. Protocols in this category include Location-Based Multicast(LBM) [6] and Voronoi Diagram based Geocasting(VDG) [7]. LBM is essentially identical to flooding packets, with the modification that a node determines whether to forward a geocast packet further via one of two schemes. In the LBM scheme 1, when a node receives a geocast packet, it forwards the packet to its neighbors if it is within a forwarding zone: otherwise, it discards the packet. A forwarding zone can be the smallest rectangle that covers both the source and the geocast region or the smallest cone covering the geocast region with the sink as the vertex. In the LBM scheme 2, whether a geocast packet should be forwarded is based on the position of the sender node at the transmission of the packet and the position of the geocast region. That is, for some parameter δ, a node B forwards a geocast packet from a node A, if the node B is at least δ closer to the center of the geocast region than the node A. The forwarding zone defined in LBM may be a partitioned network between the sink and the geocast region, although there exists a path between the source and the destination. To solve this problem, in VDG, the definition of the forwarding zone of LBM has been modified. The neighbors of the node A that are located within the forwarding zone in VDG are exactly those neighbors that are closest in the direction of the destination.

Routing-based protocols create routes from the source to the geocast region via control packets [5]. Protocols in this category include the GeoTORA [8] and Geocast Adaptive Mesh Environment for Routing(GAMER) [9] and Mesh-based Geocast Routing protocol(MGR) [10]. In GeoTORA, a source node essentially performs an anycast to any node in the geocast region via TORA which is a unicast routing protocol for MANETs. When a node in the geocast region receives a packet, it floods the packet to the geocast region. GAMER provides a mesh of paths between the sink and the geocast region. The mesh is created by flooding JOIN-DEMAND(JD) packets within a forwarding zone. Once a node in the geocast region receives a non-duplicate JD packet, it generates a JOIN-TABLE(JT) packet and unicasts it back to the source following the reverse route taken by the JD packet. All of the nodes in the reverse route become parts of the mesh. Data packets generated by the source are forwarded by the mesh members within the mesh and flooded within the geocast region. MGR is similar to GAMER.

Cluster-based protocols geographically partition a MANET into several disjoint and equally sized cellular regions and select a cluster head in each region for executing information exchange [5]. Protocols in this category include GeoGRID [11]. GeoGRID partitions the geographic area of the MANET into two-dimensional logical grids. Each grid is a square of size $d*d$. A gateway node is elected within each grid. The forwarding zone is defined by the location of the source and the geocast region and only gateway nodes in forwarding zone transmit packets. There are two schemes on how to send geocast packets: Flooding-Based GeoGRID and Ticket-Based GeoGRID.

Geographic and Energy Aware Routing(GEAR) algorithm is a geocast protocol for sensor networks [12]. It uses energy aware neighbor selection to route a packet towards the geocast region and Recursive Geographic Forwarding algorithm to disseminate the packet inside the geocast region. When a node receives a packet, among its neighbors GEAR picks the next hop minimizing the cost which is the combination of the distance to the geocast region and the consumed energy. GEAR also includes a mechanism to route around a hole.

Previous algorithms present methods for efficiently delivering geocast commands from the sink node to all the sensor nodes in a geocast region. But in many applications data sensed by the nodes in the geocast region need to be delivered to the sink node. A multicast tree built among the geocast nodes will facilitate the routing and aggregation of collected data. This multicast tree will also help deliver new commands to the nodes in the same geocast region. Zhang et al introduced algorithms to build a multicast tree for geocast nodes [13]. The tree has the sink node as the root and some non-geocast nodes as relaying nodes. All the geocast nodes are included in the bottom section of the multicast tree. They proposed 3 different algorithms for building multicast trees: Single branch Multicast tree(SAM), Cone-based Forwarding Area Multicast tree(CoFAM), and Minimum spanning tree based Single brAnch Multicast tree(MSAM). In this paper we take the approach of building a multicast tree among geocast nodes but will present more energy-efficient method for building the tree in the geocast region.

### *B. Efficient Broadcasting*

Techniques for network wide broadcasting in MANETs can be applied to broadcasting packets in a geocast region in sensor networks. In this subsection we first survey broadcasting techniques in MANETs and then introduce some protocols developed for broadcasting packets in a geocast region in sensor networks.

Broadcast techniques in MANETs are classified into four categories: simple flooding, probability based methods, area based methods, and neighbor knowledge methods [14]. The algorithm for Simple Flooding starts with a source node broadcasting a packet to all neighbors. Each of those neighbors in turn rebroadcasts the packet exactly once and this continues until all reachable network nodes have received the packet [14].

Probability based methods use some basic understanding of the network topology to assign a probability to a node to rebroadcast. There are the probabilistic scheme and the counter-based scheme in this category [15]. The probabilistic scheme is similar to flooding, except that nodes only rebroadcast with a predetermined probability. In the counter-based scheme, upon reception of a previously unseen packet, the node initiates a counter with a value of one and sets a RAD (which is randomly chosen between 0 and *Tmax* seconds). During the RAD, the counter is incremented by one for each redundant packet received. If the counter is less than a threshold value when the RAD expires, the packet is rebroadcast. Otherwise, it is simply dropped [14].

Area based methods assume nodes have common transmission distances: a node will rebroadcast only if the rebroadcast will reach sufficient additional coverage area. There are the distance-based scheme and the location-based scheme in this category [15]. In the distance-based scheme, a node compares the distance between itself and each neighbor node that has previously rebroadcast a given packet. Upon reception of a previously unseen packet, a RAD is initiated and redundant packets are cached. When the RAD expires, all source node locations are examined to see if any node is closer than a threshold distance value. If true, the node doesn't rebroadcast. In the location-based scheme, each node must have means to determine its own location, e.g., a GPS. Whenever a node originates or rebroadcasts a packet, it adds its own location to the header of the packet. When a node initially receives a packet, it notes the location of the sender and calculates the additional coverage area obtainable were it to rebroadcast. If the additional area is less than a threshold value, the node will not rebroadcast, and all future receptions of the same packet will be ignored. Otherwise, the node assigns a RAD before delivery. If the node receives a redundant packet during the RAD, it recalculates the additional coverage area and compares that value to the threshold [14].

Neighbor knowledge methods maintain state on their neighborhood, which is used in the decision to rebroadcast. There are Flooding with Self Pruning [16], Scalable Broadcast Algorithm (SBA) [17], Dominant Pruning [16], Multipoint Relaying [18], Ad Hoc Broadcast Protocol (AHBP) [19], Connected Dominating Set(CDC)-Based Broadcast Algorithm [20], and Lightweight and Efficient Network-Wide Broadcast (LENWB) [21] in this category. Among them, we describe the first two protocols. The Flooding with Self Pruning protocol requires that each node should have knowledge of its 1-hop neighbors, which is obtained via periodic "Hello" packets. A node includes its list of known neighbors in the header of each broadcast packet. A node receiving a broadcast packet compares its neighbor list to the sender's neighbor list. If the receiving node would not reach any additional nodes, it refrains from rebroadcasting: otherwise the node rebroadcasts the packet. SBA requires that all nodes have knowledge of their neighbors within a two hop radius. This neighbor knowledge coupled with the identity of the node from which a packet is received allows a receiving node to determine if it would reach additional nodes by rebroadcasting. 2-hop neighbor knowledge is achievable via periodic "Hello" packets; each "Hello" packet contains the node's identifier and the list of known neighbors. After a node receives a "Hello" packet from all its neighbors, it has 2-hop topology information centered at itself [14].

Some techniques have been proposed to efficiently broadcast packets to a geocast region in sensor networks. GEAR uses a Recursive Geographic Forwarding algorithm to disseminate the packet inside the geocast region R. Suppose that the geocast region R is the big rectangle and a node N receives a packet P for region R, and finds itself inside R. In this case, N divides the region R into 4 sub-regions, each of which is a smaller rectangle and 1/4 of R, and creates four new copies of P bound to 4 sub-regions of region R. Repeat this recursive splitting and forwarding procedure until the stop condition of recursive splitting and forwarding is satisfied. The recursive splitting terminates if the current node is the only one inside this sub-region [12]. This broadcasting technique can be applied for the geocast region with shapes which can be recursively divided but it is not desirable to make any assumptions on the shape of the geocast region.

### *C. Vulnerabilities of Geocast*

Schoch et al presented an attack model in geocast message distribution in [22]. First they described goals of an attacker in relation to Geocast. This relation is fulfilled if the attacker either a) participates in Geocat, b) interferes with Geocast, or c) interferes with layers that Geocast relies on to achieve his goals. The attacker's goals include:

1) Global denial of service: Targets the system as a whole to reduces general availability
2) Selective denial of service: Targets single nodes or single types of messages to reduce availability of the system for specific nodes or applications
3) Information flooding/displacement: Tries to inject and promote false information into the system.

They did not consider some attacker goals such as message privacy because they considered the application of Geocast in the environment of vehicular ad hoc networks. In this paper we will not consider the message privacy as a security goal, either.

Then they described attacker methods as follows:

1) Forging of messages: An attacker may create and send

messages with arbitrary content and header data, at any location, time and frequency.

2) Replay of messages: An attacker may capture messages and replay them at another location or at a later time.
3) Manipulation of messages: An attacker may modify message content or header fields like the destination region before forwarding.
4) Forwarding misbehavior: An attacker may not adhere to the forwarding rules.
5) Impersonation: An attacker may illegally assume the identity of other nodes.
6) Egoistic medium access: An attacker may not respect cooperative medium access and thus monopolize the channel.
7) Radio interference: An attacker may send jamming signals.

In this paper we propose security mechanisms to thwart the first five attack methods. We do not consider the last two attack methods because they require the cooperation with the layers 1 and 2. Instead we assume that attacks can come from not only outside attackers but also inside attackers.

## III. Basic Geocast Protocol

In this section we describe the geocast protocol in sensor networks. The protocol reduces energy consumption and facilitates the in-network data aggregation [23].

In this paper we assume that after a sensor network is deployed, a new sensor node cannot be added but a sensor node can fail to function probably due to energy exhaustion and, therefore, leave the sensor network. When a sensor network is deployed initially, all the sensor nodes broadcast a hello message to its 1-hop neighbors. The initial hello message also includes the geographic location of the sending node. After exchanging the initial hello messages, every node knows the identity and location of its 1-hop neighbors. Then every node broadcasts to its 1-hop neighbors the second hello message which includes the identities and locations of 1-hop neighbors of the sender. After exchanging the second hello messages, every node knows the identity and location of 2-hop neighbors. Then sensor nodes go into the normal operating mode, during which every node sends a hello message to its 1-hop neighbors periodically to inform its liveness and identities of failed neighbors, if any.

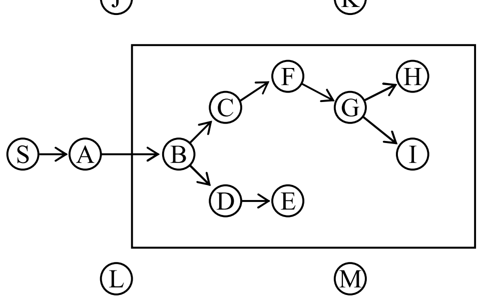

Fig. 1. Multicast Tree for Geocast

The initial geocast command issued from the sink node is routed to the first sensor node, which is called an access point (AP), in the geocast region using the GEAR protocol described in the previous section. Then using the AP node as a root, a multicast tree is built among sensor nodes in the geocast region. Fig. 1 shows a multicast tree built among nodes in the geocast region which is depicted as a rectangle. In the figure S is the sink node and B is the access point.

Now we explain the protocol which builds a multicast tree using nodes in a geocast region. A geocast node N other than the AP node receives from another geocast node M a command packet consisting of a query, a geocast region description, and a sender set. The query and the geocast region description have been originally issued from the sink. When a geocast node broadcasts the command packet to its 1-hop neighbors, it adds its sender set to the command packet. Among its 1-hop neighbors, a node selects some nodes that are invited to rebroadcast the command packet and the set of these selected nodes is called the sender set. If the command is new to N and N is in M’s sender set, N becomes M’s child in the multicast tree. Then N makes its own sender set and sends the command packet to the nodes in its sender set. N’s sender set is constructed using a greedy method. Each of N’s 1-hop geocast neighbors is checked for the inclusion in N’s sender set from the farthest node (from N) to the closest one. The node N’s n-hop geocast neighbor set is defined to be the set of all nodes which are N’s n-hop neighbors and also in the geocast region. N’s 1-hop geocast neighbor node P is included in N’s sender set, if it expands N’s 2-hop geocast neighbor set. Following is the pseudo-code description of the algorithm.

```
N receives a command packet from M;
/* command consists of query, geocast region, */
/* and M’s sender set */
if (N has seen this command or is not in the geocast region)
  return;
/* if new command and in the geocast region */
/* then become the multicast child of the sender */
become the multicast child of M and notify it to M;
if (N is not in M’s sender set)
  return;
/* N is in the geocast region and chosen to */
/* rebroadcast the command packet */
/* Now N builds its own sender set */
set SENDERS to be an empty set ;
/* SENDERS is N’s sender set */
set 2H-GN to be an empty set;
/* 2H-GN is N’s 2-hop geocast neighbor set */
set 1H-GN to be N’s 1-hop geocast neighbor set;
while (1H-GN is not empty) {
  select P from 1H-GN such that M is farthest from N;
  remove P from 1H-GN;
  if ((P’s 1-hop geocast neighbor $\cap$ 2H-GN) is not empty) {
      /* P expands N’s 2-hop geocast neighbor set */
      2H-GN = 2H-GN + P’s 1-hop geocast neighbor;
      add P to SENDERS}}
  if (SENDERS is not empty)
    broadcast (query, geocast region, SENDERS)
      to N’s 1-hop neighbors;
  /* M’s sender set is replaced with N’s sender set */
```

The AP node receives a command packet originated from the sink node and the packet consists of the query and the geocast region description. The AP node calculates its sender set using the same method as in the above algorithm, builds a new command packet by adding its sender set, and broadcasts the new command packet to its 1-hop neighbors. All the geocast nodes other than the AP node run the above algorithm when they receive the command packet. Every node in the multicast tree reports its sensor data to the sink along the multicast tree and any intermediate node can aggregate data received from its child node.

The proposed protocol first finds a route from the sink node to the AP node using energy efficient algorithm in GEAR and within the geocast region builds a multicast tree using the described broadcasting technique. Therefore, the energy consumption is reduced during the routing tree construction phase. When data is collected and reported toward the AP node, the data is delivered along the multicast tree. During the data delivery, each parent node can function as a data aggregation point. Therefore, the resulting multicast tree maximizes the in-network data aggregation among geocast nodes and reduces energy consumption during the sensor data reporting phase.

## IV. Concept of Watch Nodes

Let’s assume that two sensor nodes A and B are within each other’s communication range. When A sends a report that B is a misbehaving node, there are two possibilities. In the first, B was the misbehaving node and its anomalous behavior was detected by A. In the second, A was actually the misbehaving node and sent a false report although B had not done anything wrong. Whatever the case may be, upon receiving A’ report, B will send a refutation report. Thus other nodes which receive two conflicting report will not be able to determine which is the real misbehaving node. In this case the opinion of a third sensor node C, which has been monitoring the behavior of both A and B, is necessary. If only one node is misbehaving among A, B, and C, those nodes which have received reports from A, B, and C will find that at least two reports are consistent and be able to decide which is the misbehaving node. The node C is called a watch node for A and B. Any node which can listen to both A and B can become a watch node for A and B. Of course, the watch node C can become a misbehaving node. But even if C sends a report that either A or B is a misbehaving node, any node which receives this C’s report will disregard it because they have not heard anything from either A or B.

The more watch nodes there are, the more helpful it becomes to make correct judgment. But if there are too many watch nodes, resources of watch nodes will be wasted. To select the proper number of watch nodes, candidate watch nodes are sorted in the increasing order of the simultaneous distance from A and B and the necessary number of closest watch nodes is chosen. The simultaneous distance of a watch node C from A and B are defined as follows.

$$\text{Root}\ [\{\text{Distance}(A, C)\}^2 + \{\text{Distance}(B, C)\}^2]$$

In this paper we choose the node with the smallest simultaneous distance as a watch node to minimize the resource consumption. In Fig. 2 nodes C and D are candidate watch nodes for nodes A and B. But because C has the smaller simultaneous distance than D, C becomes the watch node for A and B.

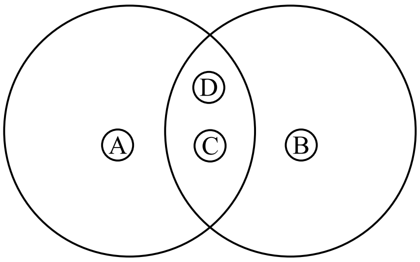

Fig2. Watch Nodes

During the initial deployment phase, every sensor node collects information on 1-hop neighbors and 2-hop neighbors and it becomes possible to determine the best watch node for any pair of nodes using this neighborhood information. The algorithm for deciding the best watch node is as follows.

```
decideWatchNode (C) {
  for every pair of nodes A &B such that. A &B are C’s 1-hop
      neighbors {
    1H-N(A) = A’s 1 hop neighbors;
    1H-N(B) = B’s 1 hop neighbots;
    1H-N(A&B) = 1H-N(A) ∩ 1H-N(B);
    if (1H-N(A&B) ∋ C) {
      calculate simultaneous distance from A & B
          for every node in 1H-N(A&B);
      if (C is the node having the smallest distance)
        C becomes the watch node for A & B}}}
```

The watch node C for A and B stores in its buffer packets which have been recently sent by A and B. If A and B send packets blaming each other for malicious behavior, C retrieves messages from A and B from its buffer, analyzes them, and decides who is the real misbehaving node. Sensor nodes A, B, and C broadcast reports on a misbehaving node with TTL value set to 2. These limited broadcast messages are guaranteed to be sent to all the 1-hop neighbors of A and B. Upon detecting a misbehaving node, neighbor nodes of a misbehaving node will cut off the communication with it and report to the sink node.

When A unicasts to B, only one watch node becomes active. But when A broadcasts to its neighbors, all the watch nodes for A and every 1-hop neighbor of A will become active.

## V. Security Extension of The Protocol

In this section we explain how the basic geocast protocol described in Section III is extended to guard against attacks inside attackers as well as outside attackers. We explain the security mechanism in three phases: initialization of a sensor network, construction of a multicast tree along with the delivery of commands, and finally delivery of collected sensor data.

### A. *Sensor Network Initialization*

To simplify the protocol we assume that there is no security attacks by malicious nodes during a short interval right after sensor nodes are deployed. During this period sensor nodes communicate trusting each other. We also assume that a sensor node N is equipped with a public key PK(N) and a private key SK(N), a GPS device, and a synchronized clock before being deployed. Upon being deployed, N senses its location using the GPS device and stores this information. At a pre-specified time after the deployment, each sensor node broadcasts its identity, location, and public key stored in a certificate format to its 1-hop neighbors via the first hello message. The second hello message is exchanged as in Section III. The above procedure can be performed during a short interval and it is not unreasonable to assume that there may be no attacks from malicious nodes.

### B. *Construction of a Multicast Tree*

To protect the multicast tree building process from malicious nodes, we modify and extend the basic geocast protocol. Before explaining detailed secure geocast protocol, we describe two features added to the basic geocast protocol as follows.

1) A node M which has received a geocast command can determine whether its location is in the geocast region or not. If a node knows that it is in the geocast region, it monitors who broadcasts the geocast command among its 1-hop neighbors and maintains the list of such nodes during some time interval. Moreover, if M's neighbor node N sends an ACK(N,L) message to its multicast parent node L to notify that N becomes a multicast child of L, M listens to this message, checks if L is a 1-hop neighbor of N, and stores this ACK message during some time interval.
2) When a node in a geocast region receives a command and decides to forward this command, it calculates its sender set. But among nodes in this sender set, some nodes may have already received the same command and forwarded it. If there exist such nodes, those nodes are excluded from the sender set. When a sender set is a null set, the node broadcasts the command with the null sender set.

With the above two additions, we explain the secure geocast protocol. When a sink node (SN) has a command to geocast, it generates the following message and sends it to its neighbor node, which is the best neighbor node on the way to the geocast region.

$\{\text{Command, TS}\}^{SK(SN)}$

where $\{M\}^{SK(N)}$ is the message appended with the signature on M generated with SN's private key and TS is a timestamp. Later this message is forwarded toward the geocast region using the following algorithm.

```
receiveCommand(signed message) {
  node M receives {command, TS}^SK(L) from node L;
  M checks integrity and freshness of message by
    checking the signature and timestamp;
  if (node M is not in the geocast region) {
    send ACK(M,L) to L;
    find the next node N toward the geocast region;
    unicast the following message to N
      {Command, TS'}^SK(M)
    if (N is not in the geocast region)
      check if N sends the command to the next node
        which is N's neighbor (otherwise broadcast attack
        report with TTL=2);
    else /* N is the first node in the geocast region */
      check if N broadcasts the same command with TTL=1
      and nodes of N's sender set are N's 1-hop neighbors
      (otherwise broadcast attack report with TTL=2);
    check if N returns ACK(N,M) to M}
  else /* M is in geocast region */
    if (has not seen this command and M is in sender set ) {
      forward the command;
      send ACK(M,L) to L}
    else if (has not this command but M isn't in sender set)
      send ACK(M,L) to L
    else if (has seen this command but M is in sender set) {
      if (N has not broadcast the command yet)
        forward the command}
    else /* has seen this command and M is not in sender set */
      no action}}
```

The command is unicast from the sink node to the AP node and broadcast within the geocast region. On the path from the sink node to the AP node, a node M delivers the command to its child node N and M watches if N properly forwards the command, illegally modifies it, or refuses to deliver it. If M reports N's attack attempt and N sends a refutation report, M and N's watch node can resolve this conflict. Any dispute regarding an ACK message that should be sent from N to M can also be resolved by the same watch node. Within the geocast region, the command is delivered by 1-hop broadcast. After M broadcasts the command, it monitors whether its receivers behave correctly. If any dispute occurs between M and its receiver, it is resolved by the corresponding watch node. But if a malicious node in a geocast region uses a false sender set, this is very difficult to detect. But because commands may be delivered by broadcasting through may different paths within a geocast region, a node M can receive the command from some other nodes with very high probability although M's parent node L happens to be a malicious node and tries to attack by using false a sender set. Moreover, nodes in a geocast region monitors if its neighbors in the geocast region send an ACK message to their parent node.

### C. *Sensor Data Delivery*

Delivery of sensor data starts from leaf nodes of the multicast tree toward the sink node. After sending data to its parent node, a sensor node should monitor whether its parent node sends upwards data which are consistent with the data that it has sent. To enable this monitoring by a child and a watch node, the message containing the data should not be encrypted but its signature is essential.

In sensor networks many sensor node can report the same data and data aggregation techniques are used to minimize the amount of data delivered in the network. Therefore, a sensor node may not forward sensor data, if it has seen the same data before and already reported upward. Moreover, a sensor node may analyze and aggregate data that it received from its many child nodes and may send upward data which are consistent with but not exactly the same as data received from the child nodes. It is easy for a node to watch if its parent node forwards data inconsistent with its data but it is difficult to decide whether its parent node maliciously ignores data that it sent. We show that this denial of service attack can be handled with watch nodes. Let's assume that a node M wants to send a data packet to its parent node N at t =T. If N had either sensed the same data or received the same data from some other child node and reported the data to its parent during the time interval <T-Δ, T>, N can ignore the same data from M. In this case M can know that N has already reported the same data upward, judges that its data report is unnecessary, and will not send the data. But if N has not sent the same data during <T-Δ, T>, M will send the data to N and N should report this data upward during <T, T+Δ>. Therefore, after sending data at t=T, M monitors N during <T, T+Δ>. If N sends consistent data upward, M stops monitoring. If N either sends inconsistent data upward or does send data upward, M reports an attack attempt by N. Although N sends a refutation packet, the watch node of M and N will help other nodes to decide that N is the malicious node. Following is the pseudo code for the above algorithm.

```
sendData(Data) {
  /* using data received from its child node L */
  /* or data sensed by itself, node M intends */
  /* to send a data packet to its parent node N at t=T */
  if (N has sent upward the same data during <T-Δ, T>)
    exit
  else {
    unicast {Data, TS}^SK(M) to N;
    if (N is not in the geocast region)
      /* M should replay up the same data during a very */
      /* short time interval */
      watch if N sends upward during short time interval
    else {
      if (N sends upward data during <T, T+Δ>)
        exit
      else if (N sends upward inconsistent data during
              <T, T+Δ>)
        report illegal data modification by N and stop timer
      else /* N does not send data upward during */
           /* <T, T+Δ> */
        report denial of service attack by N}}}
```

## VI. Conclusion

In this paper we presented an energy efficient geocat protocol for sensor networks. The protocl first finds a route from the sink node to an access point in the geocast region and then builds a multicast tree which has the access point as the root and the nodes in the geocast region as intermediate or leaf nodes in the tree. We proposed a protocol for building a multicast tree using an energy efficient broadcasting technique. The resulting multicast tree facilitates the in-network data aggregation and, therefore, saves energy during the sensor data reporting phase.

Then we extended the protocol so that it can protect commands and sensor data from security attacks from not only outside attackers but also inside attackers. After a node sends a packet to another node, it monitors whether the receiver node processes the packet correctly and reports receiver's suspicious behavior. But the problem of detecting attacks become complicated because, the detected attacker will send a refutation message to confuse neighbor nodes. The concept of a watch node is introduced to resolve this situation and, therefore, help neighboring nodes to unambiguously find the malicious node.

## References

[1] W. Zhang et al, "Energy-Aware Location-Aided Multicast Routing in Sensor Networks," in *Proceedings of WCNM*, 2005, pp.901-904.

[2] Haiyang Hu and Hua Hu, "Optimizing Energy Consumption of Data Flow in Mobile Ad Hoc Wireless Networks," *WSEAS Transactions on Computers*, vol. 7, no. 7, pp.977-987, July 2008.

[3] J. Levendovszky, A. Bojarszky, B. Karlocai, and A. Olah, "Energy Balancing by Combinatorial Optimization for Wireless Sensor Networks," *WSEAS Transactions on Communications*, 2008.

[4] P. Yao, E. Krohne, and T. Camp, "Performance Comparison of Geocast Routing Protocols for a MANET," In *Proceedings of IEEE ICCCN*, 2004, pp.213-220.

[5] J.C. Navas and T. Imielinski, "GeoCast – Geographic Addressing and Routing," In *Proceedings of MOBICOM*, 1997, pp.66-76.

[6] Y. Ko and N.H. Vaidaya, "Geocasting in Mobile Ad Hoc Networks: Location-based Multicast Algorithms," In *Proceedings of WMCSA*, 1999, pp.101-110.

[7] I. Stojmenovic, "Voronoi Diagram and Convex Hull Based Geocasting and Routing in Wireless Networks," University of Ottawa Technical Report, TR-99-11, 1999.

[8] Y. Ko and N.H. Vaidaya, "GeoTORA: A Protocol for Geocasting in Mobile Ad Hoc Networks," in *Proceedings of ICNP*, 2004, pp.213-220.

[9] T. Camp and Y. Liu, "An Adaptive Mesh-Based Protocol for Geocast Routing," *Journal of Parallel and Distributed Computing*, vol. 62, no. 2, pp.196-213, 2003.

[10] J. Boleng, T. Camp, and V. Tolety, "Mesh-based Geocast Routing Protocols in an Ad Hoc Network," in *Proceedings of IPDPS*, 2001, pp.184-193.

[11] W.H. Liao et al, "GeoGRID: A Geocasting Protocol for Mobile Ad Hoc Networks Based on GRID," *Journal of Internet Technology*, vol. 1, no. 2 2000, pp.23-32.

[12] Y. Yu, R. Govindan, and D. Estrin, "Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Netowrks," UCLA Technical Report, 2001.

[13] W. Zhang, X. Jia, C. Huang, and Y. Yang, "Energy-Aware Location-Aided Multicast Routing in Sensor Networks," In *Proceedings of International Conference on Wireless Communications, Networking, and Mobile Computing*, 2005, pp.901-904.

[14] B. Williams and T. Camp, "Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks," in *Proceedings of MOBIHOC*, 2002, pp.194-205.

[15] S. No, Y. Tseng, Y. Chen, and J. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," in *Proceedings of MOBICOM*, 1999, pp.151-162.

[16] H. Lim and C. Kim, "Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks," in *Proceedings of ACM MSWIN*, 2000.

[17] W. Peng and X. Lu, "On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks," in *Proceedings of MOBIHOC*, 2000.

[18] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint Relaying: An Efficient Technique of Flooding in Mobile Wireless Networks," INRIA Technical Report No. 3898, 2000.

[19] W. Peng and X. Lu, "AHBP: An Efficient Broadcast Protocol for Mobile Ad Hoc Networks," *Journal of Science and Technology*, 2002.

[20] W. Peng and X. Lu, "Efficient Broadcast in Mobile Ad Hoc Networks using Connected Dominating Sets," *Journal of Software*, 1999.

[21] J. Sucec and I. Marsic, "An Efficient Distributed Network-Wide Broadcast Algorithm for Mobile Ad Hoc Networks," Rutgers University CAIP Technical Report No. 248, 2000.

[22] E. Schoch, F. Kargl, T. Leinmueller, and M. Weber, "Vulnerabilities of Geocast Message Distribution," In *Proceedings of IEEE Globecom Workshops*, 2007, pp.1-8.

[23] Y.-C. Shim, "A New Geocast Protocols for a Moving Region in Sensor Networks," *WSEAS Transactions on Communications*, vol. 6, no. 4, pp.573-580, April 2007.

**Young-Chul Shim** received BS in Electronic Engineering from Seoul National University, Seoul, Korea in 1979, MS in Electrical Engineering from the Korea Advanced Institute of Science and Technology, Seoul, Korea in 1981, and Ph.D. in Computer Science from the University of California, Berkeley, California, U.S.A. in 1991. His major field of study includes Internet protocols, protocols and security in mobile and wireless communication networks, and context-awareness and security in ubiquitous computing systems.

He worked for Samsung Electronics Co. from 1981 to 1984. He joined the Department of Computer Engineering at Hongik University, Seoul, Korea in 1993 and is currently a PROFESSOR.

Prof. Shim is a member of Korean Institute of Information Scientists and Engineers, Korea Information and Communications Society, Korea Institute of Information Security and Cryptology, and Korea Information Processing Society.