A Stream Enabled Routing (SER) Protocol for Sensor Networks*

Weilian Su and Ian F. Akyildiz

Broadband and Wireless Networking Laboratory
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332

Tel: (404)-894-6616; Fax: (404) 894-7883
Email: {weilian,ian}@ee.gatech.edu

Abstract—**As the number of communication components can be integrated into a single chip increases, the possibility of high volume but low cost sensor nodes is realizable in the near future. Each sensor node can be designed to perform a single or multiple sensing operations, e.g., detecting temperature, seismic activity, object movement, and environmental pollution. As a result, a routing protocol must provide the *quality of service* (QoS) needed by the sensor nodes. A new routing protocol called *Stream Enabled Routing* (SER) is proposed to allow the sources choose the routes based on the instruction given by the sinks. It also takes into account the available energy of the sensor nodes. Also, SER allows the sink to give new instruction to the sources without setting up another path. Sources are the sensor nodes in the sensor field that are performing the sensing task. As a result, an interactive user-to-sources communication is achieved. In addition, the routing protocol is shown mathematically to perform well in the sensor network environment.**

Keywords— **Sensor Networks, Routing, Power Aware, Unicast, and Multicast.**

I. Introduction

As the number of communication components can be integrated into a single chip increases, the possibility of high volume but low cost sensor nodes is realizable in the near future. Each sensor node can be designed to perform a single or multiple sensing operations, e.g., detecting temperature, seismic activity, object movement, and environmental pollution. These sensor nodes can be used in the transportation, health care, warfare, security, and even space exploration industries. In warfare, for example, sensor nodes can be designed to detect the objects, e.g., tank, car, and human, as well as their moving directions and locations. By connecting these small nodes together by radio links, the nodes are more robust in performing sensing tasks and can provide a more precise picture of the environment than a traditional single sensor.

The unique characteristics of the sensor networks are [1]:

- Sensor nodes use broadcast communication paradigm.
- Sensor nodes are very limited in power, computational capacities, and memory.
- Sensor nodes are very prone to failures.
- The topology of sensor networks changes very frequently.
- Sensor nodes may not have global $identification$ (ID) because of the large amount of overhead.
- Sensor nodes are densely deployed in large numbers.

With these characteristics and design factors, many researchers are working toward the solutions for sensor networks. The so-called *wireless integrated network sensors* (WINS) is developed in [11], where a distributed network and Internet access are provided to the sensor nodes, controls, and processors. Since the sensor nodes are in large number, WINS networks take advantage of this short-distance between nodes to provide multihop communication and minimize power consumption. Since nodes can be very small, there may be no room for an antenna. The "Smart Dust" is developed in [9], which uses the Micro ElectroMechanical Systems (MEMS), to address this concern. These Smart Dust motes, i.e., sensor nodes, may be attached to the objects or even float in the air because of their small size and light weight. These motes may contain solar cells to collect energy during the day, but the drawback of the Smart Dust motes is that they require a line of sight to communicate optically with the base-station transceiver.

A family of adaptive protocols called *Sensor Protocols for Information via Negotiation* (SPIN) [5] is designed to address the deficiencies of *classic flooding* by *negotiation* and *resource-adaptation*. SPIN has three types of

*** To Appear in "Med-hoc-Net 2002," Sardegna, Italy, September 2002.**

messages, i.e., ADV, REQ, and DATA. Before sending a DATA message, the sensor node broadcasts an ADV message containing a descriptor of the DATA. If neighbors do not have the data, they send a REQ message for the DATA. This type of protocol is good for disseminating information to all sensor nodes. Yet, it cannot isolate the nodes that do not want to receive the information. As a result, unnecessary power may be consumed.

Also, a *directed diffusion* data dissemination paradigm is proposed in [7]. The sink sends out $interest$, which is a task description, to all sensor nodes. The task descriptors are named by assigning attribute-value pairs that describe the task. If the sources do have data for that $interest$, the data is routed along the reverse path of interest propagation. The interest, data propagation, and data aggregation are determined locally. The sink has to refresh and reinforce the interest when it starts to receive data from the sources. However, this approach does not address the *quality of service* (QoS) needed by the connection between the source and sink, such as delivering data in the shortest time, power aware of the selected route, or the ability to change interest for the selected sources without rebroadcasting a new interest to search for the sources again.

The use of power-aware metrics in making routing decision to prolong an ad-hoc networks' life-time and its time to node failure is addressed by [13]. Such metrics are useful for sensor networks, but sensor nodes are lower in battery, lesser in computational capabilities, and lower in memory than the nodes in the ad-hoc networks. Also, sensor nodes lack global IDs, such as *Internet Protocol* (IP) addresses. As a result, a routing protocol for sensor networks has to take into account of these differences.

Since sensor nodes require QoS regardless of their environmental and technical constraints, we propose a new routing protocol called *Stream Enabled Routing* (SER). The routing protocol requires the sinks to specify the sensor nodes that perform the tasks in their instructions. If the nodes do not have a *global positioning system* (GPS), then they can use a location awareness protocol, such as [12], to approximate their locations. SER can be integrated with the application layer very easily, because it is based on instructions or tasks. Instead of assigning attributes to a task as in [7], an instruction is predefined as an identifier value. This way only the identifier is sent and not the whole attribute list in order to conserve memory. There are four types of messages, i.e., *scout message* (*S-message*), *information message* (*I-message*), *neighbor-neighbor message* (*N-message*), and *update message* (*U-message*). The *S-message* is broadcast, so the sources can select the routes between the sources and sinks based on the QoS requirements of the instruction. The routing protocol takes into account the available energy of the sensor nodes, the QoS requirements of the instruction, the memory limitation of the nodes, and the localized effect of the heavily dense nodes. After the route is established, it allows the sink to give new instructions to the sources without setting up another route. This dynamic setup of routes has the following benefits when compared to traditional routing protocols [10] [4], SPIN [5], and *directed diffusion* [7].

- Periodic update of the routes is not needed in order to conserve energy.
- It is able to adapt to failures.
- It is also able to cope with topology changes.
- A routing table is not needed at each sensor node. As a result, memory usage is minimized at each node.
- It can easily incorporate new sensor nodes into the route selection process.
- Sources determine the routes based on QoS requirements.
- It allows one-to-one, many-to-one, one-to-many, and many-to-many communications.
- It exploits the benefits of topology maintenance protocols, e.g., SPAN [3], GAF [15], and LEACH [6].

In Section II, we present the new routing protocol SER. In Section III, we provide a mathematical analysis of SER and clustering based techniques to investigate the power consumption and emission. We also perform simulations of SER in Section IV. In Section V, we conclude the paper.

II. Stream Enabled Routing (SER) Protocol

A. Overview

The SER protocol consists of seven phases:

- Phase I: *Source Discovery*.
- Phase II: *Route Selection*.
- Phase III: *Route Establishment*.
- Phase IV: *Route Reconnection*.
- Phase V: *I-message Transmission*.
- Phase VI: *Instruction Update*.
- Phase VII: *Task Termination*.

The *S-message* is used during Phase I as shown in Figure 1.(a) to find the sources that will carry out the instruction specified in the *S-message*. Once the sources are found, the sources decide the type and level of the routes needed by the instruction. There are four types of route, each with two levels, i.e., Level-1 and Level-2. The different levels are depicted in Figure 1.(b). The value μ is the radius of Level-2 routes. The combination of type and level of routes gives rise to a new concept called $stream$. A typical hop-to-hop route, which involves only one node to another to form a route, is a stream at Level-2 with $\mu = 0$, i.e., this is also the Level-1 stream. Note that each Level-2 stream has a Level-1 stream as well. At Level-2, the radius μ of the stream can increase as large as needed to satisfy the QoS specified by the instruction. While Level-1 uses one-to-one communication, Level-2 uses modified flooding with data flowing only toward the sink through the stream. Also, the combination of types and levels gives different levels of QoS to a stream.

After the streams are selected, the source sends an *N-message* to establish the streams back to the sink as shown in Figure 1.(c). If the streams are disconnected due to node failure and/or low-energy level, the SER protocol repairs them by using *N-message* and *S-message*. Once the streams are established, data start to flow from the sources to the sink through either Level-1 or Level-2 streams with *I-message*. The sink can update the task at the sources through either Level-1 or Level-2 streams by using the *U-message* depending if Level-1 or Level-2 streams are selected to route the data. Both sink and sources can also terminate the streams by the *U-message* as shown in Figure 1.(d).

B. Source Discovery

We define a *sensor field* as an area, which the sensor nodes are being deployed. Since the topology of the sensor network changes frequently and the sensor nodes fail quite often due to low energy level or interferences, the routes from the sink to the sources should be set up dynamically when sensed information, i.e., descriptors, are requested from the sources. For example, if sensor nodes are asked to detect temperature, the descriptor is the temperature value. If nodes are asked to detect the type of animals, the descriptor is a number that is mapped to the type of animals. Also, if sensor nodes are asked to take a picture of the environment, the descriptor can be the whole or part of the image of the environment snapshot.

A sink broadcasts a short *S-message* to mark the possible routes from the sink to the sources. The fields of the *S-message* are illustrated in Figure 2. The TID field is the *task ID* field, which consists of four subfields, i.e., LI, MT, INS, and $TLOC$ as shown in Figure 3. The *length indicator* (LI) indicates the length of the message. The *message type* (MT) field indicates the type of message that this packet is carrying, i.e., MT=0 stands for *S-message*; MT=1 indicates an *I-message*; MT=2 represents an *U-message*; and MT=3 corresponds to a *N-message*. The $instruction$ (INS) subfield maps a numeric value to a specific instruction, and the $TLOC$ subfield represents the *targeted location*. For example, the sink gives the instruction "Sensor nodes detect temperature at every 10 minutes in 10 meters radius", and this instruction may be mapped to an INS value of 0. The instruction tells the sensor nodes that are within the radius of 10 meters from the location specified by the $TLOC$ field to detect the temperature at every 10 minutes. Since each node is designed to perform a specific task, e.g., detecting temperature, the number of instructions may be very small, and the INS values representing the instructions may be predefined and loaded into the nodes initially.

TID	*NAP*	*LID*	*NH*	*AE*

TID = Task ID *NAP = Network Access Point*
LID = Local ID *AE = Average Energy of the Route*
NH = Number of Hops From the Sink

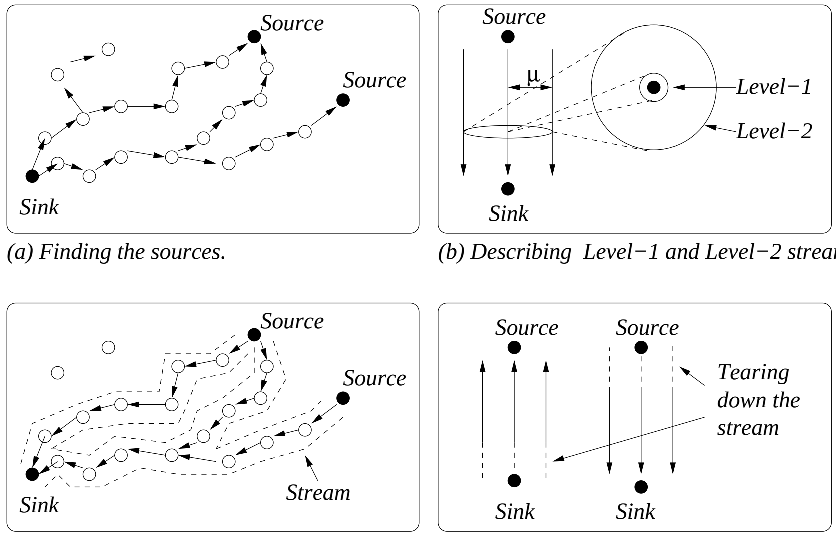
Fig. 2. S-message.

To indicate where the instruction is originated, the *network access point* (NAP) field contains a value, which represents a unique sink. The number of sinks deployed is very small when compared to the number of nodes in the network. For example, there maybe only 3 or 4 sinks when 4 to 5 thousands sensor nodes are in the sensor field. Since the *S-message* is routed to all sensor nodes in the sensor field, a node must be able to determine the neighbor that has sent the message. Each node in the sensor field has a *local ID* (LID) that is selected randomly from a set, which has values ranging from 1 to κ, where κ is the maximum value of the set. The total number of nodes of which the *S-message* has been received prior to the current sensor node is captured by the number of hops (NH) field as shown in Figure 2. There is also an *average energy* (AE) field whose value is computed by equation (1), which is the average energy of the route that the *S-message* has traversed prior to the current node.

$$AE = \frac{NH_{i-1} \cdot AE_{i-1} + E_i}{NH_{i-1} + 1} \tag{1}$$

where NH_{i-1} and AE_{i-1} are the values stored in the NH and AE fields of the received scout message at the $(i-1)^{th}$ sensor node, and E_i is the available energy at the i^{th} node. The subscript i represents the previous node that has received the *S-message* prior to the current sensor node.

Whenever a node receives an *S-message*, it checks to see if the instruction, i.e., INS, is intended for it. If the instruction of the *S-message* is not intended for the node, the node stores the TID, NAP, LID, NH, and AE values in a *connection tree* (*C-tree*), which is a logical tree that represents the possible connections through the node. Hence, the *C-tree* keeps track of the node's neighbors that are capable of routing information back to the sink. The *C-tree* has the following tree structure as shown in Figure 4. The Γ node contains the INS and $TLOC$ values of the TID field. The NAP, $DLID$, $ULID$, *Downlink Sensor Problem* (DSP), and *Node Selected* values are stored



(a) Finding the sources.
(b) Describing Level−1 and Level−2 stream.
(c) Routing I−message to sink.
(d) Tearing down the stream.

Fig. 1. Overview of SER protocol.

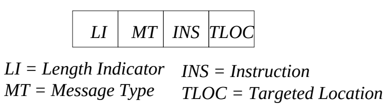


Fig. 3. TID field.

in the Ψ node, and the Φ node contains the LID, AE, and NH values. The $DLID$ value is used to store the LID value of the neighbor sensor node that will route the *I-message* back to the sink, which is also the downlink sensor node. The $ULID$ is the LID of the uplink sensor node, so an *U-message* can be forwarded to sources from the sink or route reconnection is possible by using the *N-message*. As a result, a sensor node in an established route knows the LID values of the uplink and downlink sensor nodes. Initially, the $DLID$ and $ULID$ are not set. The DSP indicator is used to indicate if the downlink sensor node is having a problem in routing the *I-message*. The *Node Selected* is used to indicate if the node is selected for routing. Initially, both DSP and *Node Selected* are set to OFF. The contents of the Γ, Ψ, and Φ nodes in the *C-tree* are summarized in Figure 5.

After the node stores the values, it calculates a new AE by equation (1) with i incremented by 1. In addition, it increases the value stored in the NH field shown in Figure 2 by 1. The sensor node then inserts the new AE, new NH, and LID of the node into the AE, NH, and LID fields of the *S-message*, respectively, and then broadcasts the updated *S-message* to its neighbors. If the sensor node happens to receive the same *S-message* from its neighbors, it does not do anything. As a result, the *C-tree* has only one Φ node.

After the sources receive the first *S-message*, they keep on listening for *S-messages* with different LID but with the same TID and NAP fields for σ seconds. The sources store the TID, NAP, LID, NH, and AE values in a

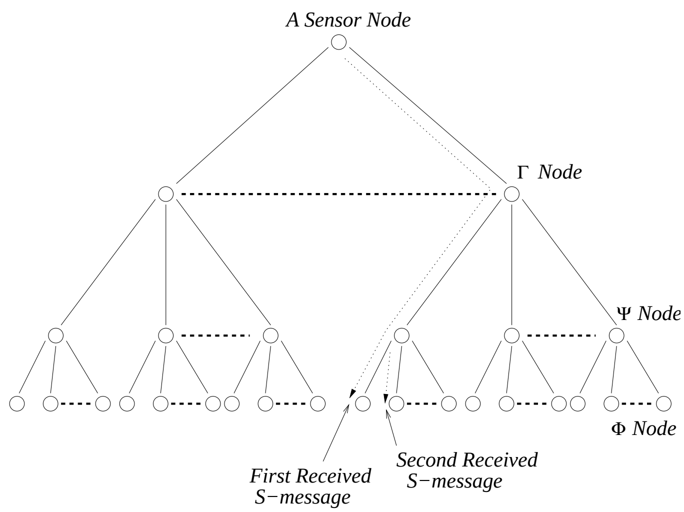


Fig. 4. Logical tree structure.

Node \ Tree	C−tree	T−tree
Γ Node	INS and TLOC	INS and TLOC
Ψ Node	NAP, DLID, ULID, DSP, and Node Selected	NAP, $DLID_1$ ... $DLID_\chi$, and DSP_1 ... DSP_χ
Φ Node	LID, AE, and NH	LID, AE, NH, and τ_j

Fig. 5. Contents in the Γ, Ψ, and Φ nodes of *C-Tree* and *T-tree*.

TID	NAP	LID	SLID	MES

TID = Task ID *LID = Local ID*
NAP = Network Access Point *SLID = Selected ID*
MES = Message

Fig. 6. N-message.

task tree (*T-tree*), which also has the same tree structure as shown in Figure 4. Unlike the *C-tree*, the *T-tree* can have more than one Φ node. The *T-tree* is to hold information related to the task being assigned to it. Instead of just one $DLID$ value stored in the Ψ node, the *T-tree* contains χ $DLID$ values, because each of the sources can select upto χ LIDs to route the *I-message* back to the sink depending on the QoS requirements. The maximum value of χ is the number of neighbor nodes. For each $DLID$ value, there is also a DSP indicator in the Ψ node. On the other hand, the Ψ node has no $ULID$ value, i.e., the LID of an uplink sensor node, and *Node Selected* indicator, because the sources are the destination of the *S-message*. The Φ node of the *T-tree* also contains the *arrival time* (τ_j) of the *S-message*, where j represents the j^{th} received *S-message*. A source can receive a maximum of χ *S-message* since it has χ neighbors. The route associated with the first received *S-message* is considered the shortest route while the route associated with the last received *S-message* is the longest route. The contents of the nodes in the *T-tree* are summarized in Figure 5. After σ seconds, the sources select the neighbor sensor nodes, i.e., the LIDs of the neighbor nodes, for transmitting the *I-message* back to the sink according to the QoS requirements of INS.

C. Route Selection

After the sources have received the *S-message*, they will determine the QoS required for the task being assigned by the *S-message*. There are 4 types of stream, which the sources can establish and communicate with the sink, and each stream can either be at Level-1 or Level-2. Below is a list of the types and their associated action carried out by the sources.

1. **Type 1: Time Critical But Not Data Critical:**
   Action: Choose LID with the lowest τ value, i.e., τ_1.
2. **Type 2: Data Critical But Not Time Critical:**
   Action: Choose χ LIDs with the highest AE.
3. **Type 3: Not Time and Data Critical:**
   Action: Choose the LID with the highest AE.
4. **Type 4: Data and Time Critical:**
   Action: Choose χ LIDs with the lowest τ, i.e., τ_1 ... τ_χ.

Note that priority is given to a LID value of a neighbor node if the LID value is contained in more than one Φ node of the same Γ node as shown in Figure 4 for types 2 and 3 streams. This way data can be aggregated if they are the same. After the neighbor nodes have been selected by the sources, the sources broadcast an *N-message* to their neighbors indicating the level and size of the stream. The fields of an *N-message* are specified in Figure 6.

If the stream is chosen to be at Level-1, the width of the stream is set to 0, i.e., $\mu = 0$. At Level-1, the messages are routed back to the sink via hop-by-hop communication, i.e., the messages are sent only to one node. The different scenarios of streams flowing between the sources and sinks are illustrated in Figure 7. There are only one source and one sink for the stream formed by Figure 7.(a). If there are more than one source, the streams can joined together if they meet somewhere between the sources and the sink as shown in Figure 7.(b). The streams can also diverge to multiple sinks if the messages are intended for multiple sinks. The streams shown in Figure 7 are Level-1 streams where nodes communicate with only one node in either the downlink or uplink direction. A Level-2 stream is formed when the size of the stream μ is greater than 0. The Level-2 stream also consists of the Level-1 stream as shown in Figure 8. The Level-1 stream will serve as the backbone in setting up Level-2 stream. The value μ is the number of hops away from the nodes in the Level-1 stream. Once the Level-2 stream is established, messages can flow downhill to the sink or uphill to the sources by flooding. Only the sensor nodes that are part of the stream participate in the flooding process. The *I-message* flows downhill by using the NH value stored in the Φ node of the *C-tree* in each sensor node as the potential. The nodes nearer the sources have higher NH values while the sensor nodes nearer the sink have lower NH values. On the other hand, the *U-message* from the sink to the sources flows uphill by using the negative of the NH values as the potential. As a result, the nodes nearer the sources have higher negative values. The flow concepts are illustrated in Figure 9 with H_{max} indicating the maximum number of hops from the sink to the source. The different types of stream with level combination are presented in Table I. The stream $S(2,2)_{\chi,\mu}$ is of type 2 at Level-2 with μ stream width and χ neighbors routing the messages.

	Type 1	**Type 2**	**Type 3**	**Type 4**
Level-1	$S(1,1)_{1,0}$	$S(2,1)_{\chi,0}$	$S(3,1)_{1,0}$	$S(4,1)_{\chi,0}$
Level-2	$S(1,2)_{1,\mu}$	$S(2,2)_{\chi,\mu}$	$S(3,2)_{1,\mu}$	$S(4,2)_{\chi,\mu}$

TABLE I
DIFFERENT COMBINATION OF STREAMS ($\mu > 0$ FOR LEVEL-2).

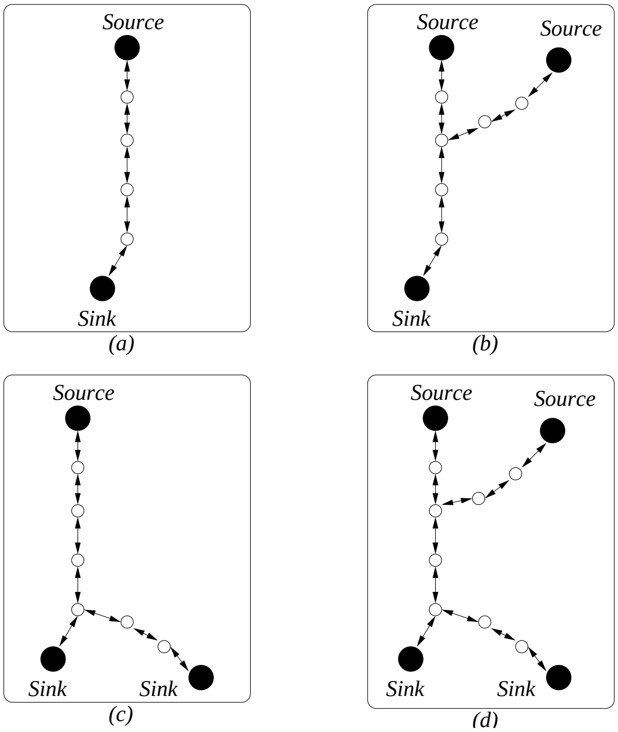


Fig. 7. Different scenarios of streams: (a) single source and sink, (b) multiple sources and single sink, (c) multiple sink and single source, and (d) multiple sources and multiple sinks.

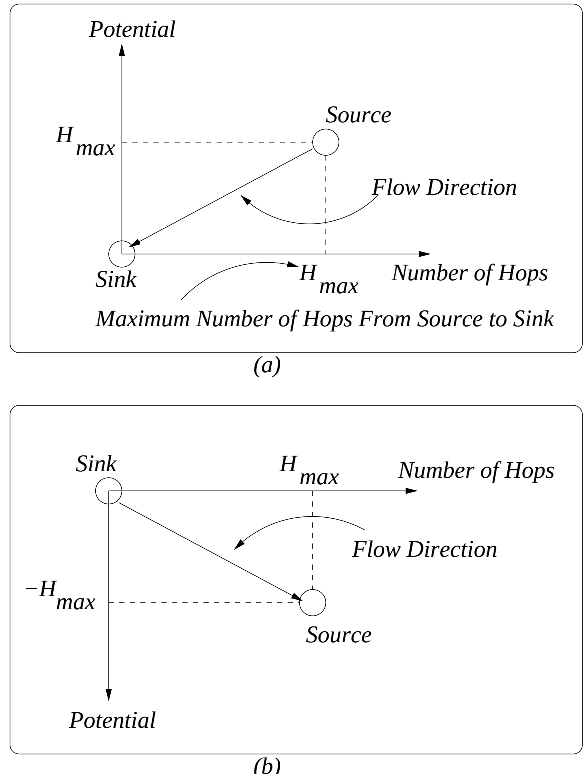


Fig. 9. Stream flow concepts: (a) downhill flow and (b) uphill flow.

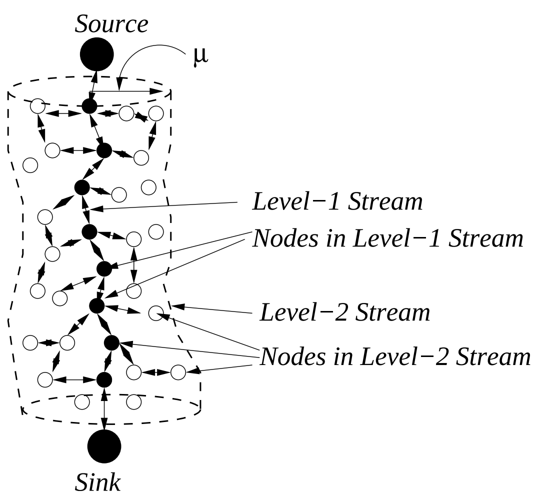


Fig. 8. Level-2 stream.

D. Route Establishment

A sensor node uses the *N-message* to tell neighbors about its local information. Once the source has decided on which neighbor sensor nodes to carry its *I-messages*, it sets the $DLID$ values, which are stored in the Ψ node, equal to the LIDs chosen according to the QoS requirements of the assigned task. After which, it sends an *N-message* as shown in Figure 6 with MES, i.e., the $message$ field, set and mapped to an *new connection message with value* μ indicator. If μ is equal to 0, the stream is at Level-1, or otherwise, it is at Level-2. The INS, $TLOC$, and NAP values of the *N-message* are the same as the *S-message*'s. The LID field of the *N-message* is set equal to the LID value of the broadcasting sensor node, and the *Selected LID* ($SLID$) value is set equal to the $DLID$ value stored in the Ψ node of the *T-tree* at the source. If there are χ $DLID$ values chosen, then χ *N-messages* are broadcasted by the source.

After the broadcast, the neighbor nodes receive and check if the TID and NAP values match the ones in the *C-tree*. If a match is found, the nodes extract and compare the $SLID$ value in the *N-message* with their LID value. If the $SLID$ value does match the LID value of the nodes, the nodes set the $DLID$ value in the Ψ node of the *C-tree* equal to the LID value of the Φ node. The nodes also set the $ULID$ value in the Ψ node equal to the value stored in the LID field of the *N-message*. In addition, the *Node Selected* indicator is also set to ON. The nodes then broadcast a new *N-message* with LID and $SLID$ values set equal to the LID and $DLID$ values of the sensor nodes, respectively. The MES value in the *N-message* stays the same as the one that is received.

If the $SLID$ value does not match the LID value of the sensor nodes, but the TID and NAP values do match the ones in the *C-tree*, and the value μ specified by the MES

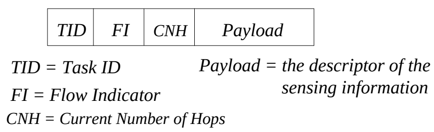


Fig. 10. I-message.

field of the *N-message* is greater than 0, the sensor nodes know that a Level-2 stream is requested; the nodes then set the *Node Selected* indicator to ON in the Ψ node of the *C-tree*. These sensor nodes rebroadcast the *N-message* with $SLID$ set equal to 0 and μ value decreased by 1. Sensor nodes receiving the same *N-message* but with different μ value do not rebroadcast. They only rebroadcast when they first receive the *N-message*. When the μ value is decreased to 0, the sensor nodes stop the rebroadcast of the *N-message*. As a result, only sensor nodes that are μ hops away from the nodes in the Level-1 stream participate in the Level-2 stream. Note that a node, which is part of the Level-1 stream, rebroadcasts the *N-message* when it receives the *N-message* from either a Level-1 or Level-2 node. Hence, a Level-1 node may rebroadcast twice while a Level-2 node only rebroadcasts once.

If the $SLID$ value does not match the LID value of the sensor nodes and the μ value in the *N-message* is equal to 0, the nodes delete the tree branch beginning at the Ψ node that has the same value as the value stored in the NAP field of the *N-message*. As a result, the Ψ and Φ nodes are deleted. If a Γ node has no Ψ node connecting to it, the Γ node is also discarded. As a result, sensor nodes that are not part of a stream remove all the data that are associated with the *N-message* from the *C-tree*. All intermediate sensor nodes between the source and sink perform these tasks.

If all intermediate nodes between the sources and sinks have not received an *N-message* in response to a *S-message* in ζ seconds, the sensor nodes delete the tree branch, which is associated to the *S-message*, from the *C-tree*. The nodes free up the memory, so they can store other incoming *S-messages*. The value of ζ can be equal to a couple of seconds depending on how large is the *sensor field*.

Once the *N-message* has reached the sink, the minimum delay or the maximum average energy stream is established. At this point, the source can send *I-messages* to the sink. Note that one-to-one, many-to-one, one-to-many, and many-to-many streams can also be established allowing unicast and multicast communications.

E. I-message Transmission

Once the sources have broadcasted the *N-message* with MES indicating an *new connection message with value*

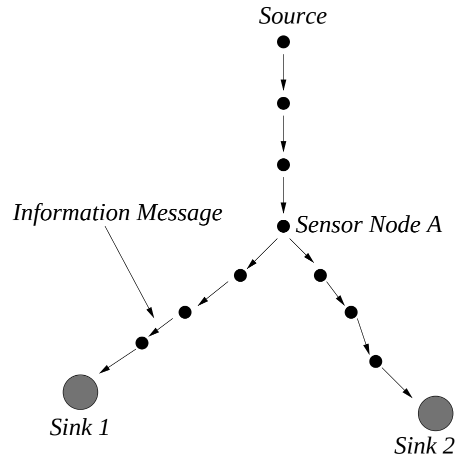


Fig. 11. First part of the streams shared.

μ, they can start sending *I-messages*. The fields of the *I-message* are illustrated in Figure 10. The TID field contains the instruction, i.e., the same INS and $TLOC$ fields as the *S-message*, that is given by the sink, so neighbor sensor nodes can determine if they are responsible to route the *I-message*. The FI field is only 1 bit long, which is used to indicate if the message is going uphill (FI=1) or downhill (FI = 0). The CNH field contains the NH value stored in the Φ node of the *T-tree* or *C-tree* of the broadcasting node. When the source broadcasts the *I-message*, it sets the CNH field with the value from the *T-tree*. The intermediate nodes between the source and sink use the *C-tree*. Also, the $Payload$ field of the *I-message* contains the descriptor of the sensing information.

Note that only the TID field is needed by the neighbor nodes to determine if they are responsible to route the *I-message*, because each of the neighbor nodes maintain a *C-tree*. The values in the FI and CNH fields of the *I-message* are only used when the stream is at Level-2, so that the *I-message* can flow downhill toward the sink via flooding using the potential as described in Section II.C. Each node only rebroadcasts once to avoid a node from rebroadcasting the same message again and again. After a sensor node receives an *I-message*, it turns OFF the receiver for an amount of time if the sleep mode operation is ON; otherwise, the receiver stays ON. The reason for turning OFF the receiver is to avoid listening to neighbors broadcasting the same *I-message*, which the node is not interested. The *C-tree* indicates which instructions the sensor nodes have to route, and the ones that are not allowed are not stored in the *C-tree*. As a result, only one copy of the descriptor is sent by the source if the descriptor is intended for different sinks and the first part of the streams selected for different sinks is the same as shown in Figure 11. The neighbor nodes at the downlink of sensor node A

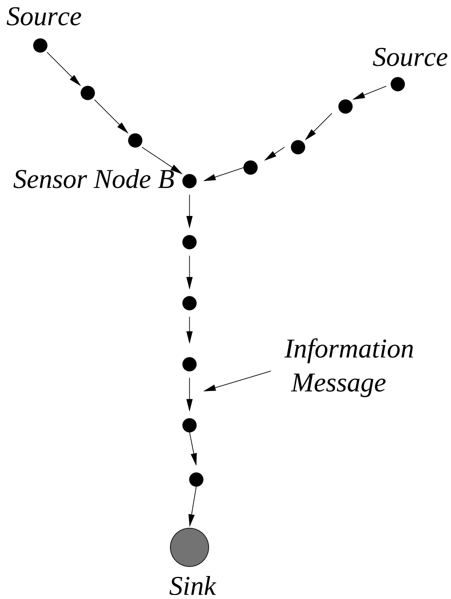


Fig. 12. Bottom part of the streams shared.

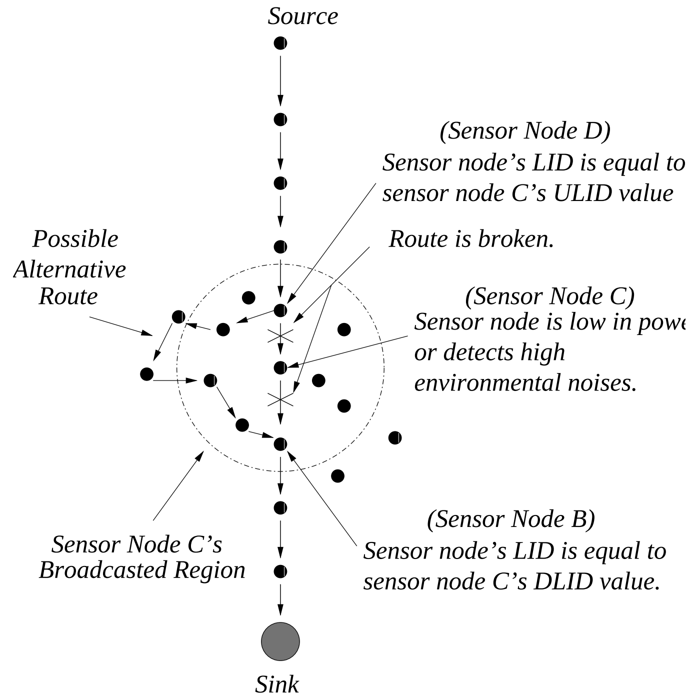


Fig. 13. Reconnecting a stream.

route the *I-message* along their own path once sensor node A has broadcasted it. If multiple sources send the same *I-messages* back to the sink and their streams are shared as illustrated in Figure 12, then only one *I-message* has to be sent from sensor node B to the sink. As a result, the nodes should have a small buffer to store incoming messages, so nodes can compare the *I-messages* and avoid unnecessary power dissipation if the messages are the same.

F. Route Reconnection

During transmission of information at Level-1 from the source to the sink, a sensor node may determine that it is low in energy for routing or there are high environmental noises around the node. After such decision, the node broadcasts an *N-message* with the MES field set equal to a value representing the *reconnect message* indicator. For this kind of *N-message*, the LID and $SLID$ fields are set equal to the sensor node's stored $ULID$ and $DLID$ values from the Ψ node of the *C-tree*, respectively.

Once the neighbors have received the *N-message*, they check their *C-trees* and determine if they have the same INS, $TLOC$, and NAP values as in the *N-message*. If they have the same values, the neighbor nodes whose LID values are the same as the values in the LID and $SLID$ fields of the *N-message* are the uplink and downlink sensor nodes, respectively, of the selected Level-1 stream as shown in Figure 13. The uplink and downlink nodes are sensor nodes D and B, respectively, as shown in Figure 13. The neighbor nodes do not rebroadcast this kind of *N-message*, because it is intended for sensor nodes D and B. Sensor node D turns the DSP indicator ON in the Ψ node of the *C-tree* and wait for an *N-message* from sensor node B. The sensor node B broadcasts a new *N-message* with the LID and $SLID$ fields set equal to the sensor node B's LID and sensor node D's LID values, respectively. The MES field is set equal to the *reconnect the route* indicator.

This new *N-message* from sensor node B is rebroadcasted by its neighbor nodes with the value in the LID field replaced by the LID value of the neighbor nodes. The nodes that have received this *N-message* create a new branch in the *C-tree* with the values in the TID, NAP, and LID fields in the same way as if they have received a *S-message* except that the average energy and the number of hops from the sink are not calculated and used.

Once a sensor node receives this *N-message*, it also checks to see if it is the uplink sensor node specified by the *N-message*. To be the uplink sensor node, the LID value of the node must be the same as the value in the $SLID$ field and the DSP indicator in the Ψ node of the associated instruction must be ON. After the sensor node D has received this *N-message*, it updates the LID value in the Φ node of the associated instruction in the *C-tree* with the value in the LID field of the *N-message*. It also turn the DSP indicator OFF. Note that if sensor node D receives more than one copy of the same *N-message*, sensor node D uses the first received *N-message*, which is also the route that has the minimum routing time. As a result, sensor node D can route the *I-message* to the neighbor node whose LID value is the same as the updated LID value.

Before routing the *I-message*, sensor node D broadcasts a new *N-message* with LID and $SLID$ fields set equal to the LID and updated LID values stored in the Φ node of the *C-tree*, respectively. Sensor node D sets and maps the MES value to an *new connection message with* μ *= 0* indicator. The neighbor nodes check if they are selected in the same way as described in Section II.D. After sensor node B has received the *N-message*, the stream is reconnected between sensor node D and sensor node B as shown in Figure 13, and sensor node B does not rebroadcast the *N-message*.

G. Route Experienced Sudden Death

There is also another scenario which affects the routing of *I-message* from the sources to the sink. Such scenario is when the stream suddenly terminates, i.e., sudden death. If the sink does not get the *I-message* at the time when it expects, the sink sends out a new *S-message* with a higher QoS requirements version of the same instruction, i.e., a higher QoS INS value. By doing this, new streams can be established to avoid trouble spots experienced by the stream which suddenly terminates. Also, if the instruction previously requires only one stream to be established, multiple streams at Level-2 can be established because the QoS requirements are stricter than before. Note that if the environment is known to inflict sudden death easily, the QoS requirements of the instruction should be stricter at the beginning. As a result, multiple streams at Level-2 can be set up between the sources and sink to enhance the robustness of the *I-message* routing.

H. Instruction Update

The last type of messages is the *U-message*. The *U-message* allows the sink to update its instruction to the sources. From the previous example, ”Sensor nodes detect temperature at every 10 minutes in 10 meters radius” can be updated to ”Sensor nodes detect temperature at every 1 minutes in 10 meters radius”. The fields of the *U-message* are shown in Figure 14. It contains the TID, FI, CNH, and $NINS$ fields. The INS and $TLOC$ subfields of the TID field are the same as the ones used by the *S-message* to establish the stream at the beginning. The FI field is used to indicate if the message is going uphill or downhill; it serves the same purpose as the FI field of the *I-message*. The CNH field contains the NH value stored in the Φ node of the $C-tree$, which is associated with the instruction specified in the TID field, of the broadcasting node. The $NINS$ contains the new instruction for the sources. The *U-message* from the sink to the sources flows uphill while it flows downhill from the sources to the sink when the streams are at Level-2 as described in Section II.C.

TID	*FI*	*CNH*	*NINS*

TID = Task ID   FI = Flow Indicator
CNH = Current Number of Hops
NINS = New Instruction

Fig. 14. U-message.

I. Task Termination

There are two situations when a task at the sources are terminated. The first situation is when the sources have finish the task associated with the instruction given by the sink. The sources broadcast a *U-message* with $NINS$ field set and mapped to a *task completed instruction* indicator. As this *U-message* is routed to the sink, the streams are teared down by removing the tree branch associated with this instruction in the *C-tree* at the intermediate sensor nodes and the *T-tree* at the sources.

The second situation is when the sink decides to terminate the instruction. The sink sends a *U-message* with $NINS$ value set and mapped to a *termination instruction* indicator. The streams to the sources are teared down as the *U-message* is routed.

III. Mathematical Analysis

A. Transmission Power

The power P_s at the receiver in wireless communication [8] is:

$$P_s = \frac{P_t g_t}{4\pi R^2}\frac{g_r \lambda^2}{4\pi} \ \ (watts) \tag{2}$$

where P_t is the output power at the transmitter; g_t is the receiver antenna gain; g_r is the transmitter antenna gain; λ is the wavelength of the transmitted signal; and R is the distance of transmission in meters (m). In sensor network communication, the attenuation of the transmitted signal can be as high as the 4^{th} order exponent of the distance R [11], because the sensor nodes are very near the ground. As a result, P_s at a sensor node is further attenuated, and the new value is given by P_r.

$$P_r = \frac{P_s}{\alpha R^{\varphi}} \tag{3}$$

where P_s is given by equation (2); φ ranges from 0 to 2; and α is the additional attenuation constant for the sensor network environment that has units of $m^{-\varphi}$. P_r can be rearranged and represented as follows:

$$P_r = P_t g_t g_r \left[\left(\frac{\lambda}{4\pi}\right)^2 \cdot \frac{1}{\alpha R^k}\right] \tag{4}$$

where P_t, g_t, g_r, λ, α, and R are the same as described in equations (2) and (3); and k ranges from 2 to 4. The right

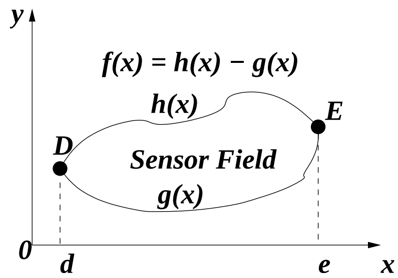


Fig. 15. Area of sensor field.

most term of equation (4) is the free space path loss or free space attenuation ℓ, which is calculated by equation (5).

$$\ell = \left(\frac{\lambda}{4\pi}\right)^2 \cdot \frac{1}{\alpha R^k} \tag{5}$$

where λ, α, R, and k are the same as described in equation (4). The free space attenuation ℓ expressed in decibels (dB) as a positive quantity is:

$$L(dB) = 10log\alpha + 10 \cdot k \cdot logR - 20\left[log\lambda - log(4\pi)\right] \tag{6}$$

Assuming the transmitters and receivers are isotropic (g_r=g_t=1), the required transmission power P_t is obtained by rearranging the terms in equation (4) and substituting the values of g_r and g_t with 1.

$$P_t = \alpha P_r \left(\frac{4\pi}{\lambda}\right)^2 R^k \tag{7}$$

where α, P_r, λ, R, and k are the same as described in equation (4).

B. Representation of a Sensor Field

The area of a sensor field given in Figure 15 can be calculated by the following equation:

$$A = \int_d^e f(x)dx \tag{8}$$

where $f(x)$ is the function that describes the sensor field between points D and E as shown in Figure 15. In addition, $f(x)$ is the difference between $h(x)$ and $g(x)$.

The sensor field can also be represented as if composed of many squares with each having an area of $\delta\ m^2$. The density of the sensor nodes ξ in each square is:

$$\xi = n\ nodes/\delta\ m^2 \tag{9}$$

If the sensor nodes are randomly distributed, the number of nodes lies on the horizontal axis is $\sqrt{n}$ and the distance d between two nodes as shown in Figure 16 is:

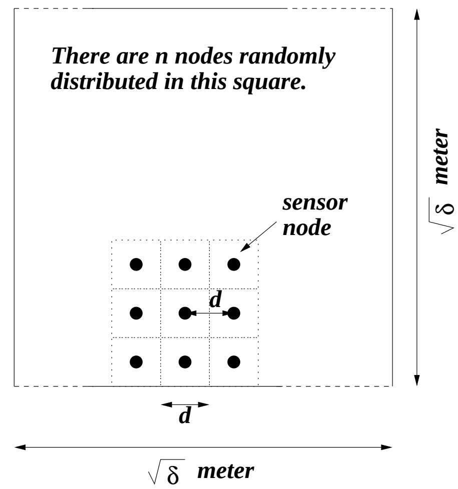


Fig. 16. One square with area $\delta\ m^2$.

$$d = \sqrt{\frac{\delta}{n}} \tag{10}$$

The dimension of the square in Figure 16 is $\sqrt{\delta}\ m$ by $\sqrt{\delta}$ m, and there are n nodes in the square.

Note that the number of randomly distributed nodes within radius R [2] is:

$$\phi = \left(N\pi R^2\right)/A \tag{11}$$

where N is the number of nodes randomly distributed in the sensor field; R is the distance of transmission; and A is the area given by equation (8). If the area of the square shown in Figure 16 is small, ϕ is approximately equal to n when R has a value of $\sqrt{\frac{\delta}{2}}$.

C. Power Consumption Based on Clustering Techniques

The sink broadcasts the task at distance q away from the cluster head z as illustrated in Figure 17. The value of q can be calculated as follows:

$$q = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} \tag{12}$$

where (x_0,y_0) and (x_1,y_1) are the location of the sink and source nodes, respectively. After the cluster head receives the task, it broadcasts to all the nodes within radius $r = \sqrt{\frac{\delta}{2}}$ as shown in Figure 18. The power $P_{c,t}$ to broadcast to all the sensor nodes within radius r from node z as shown in Figure 18 is determined by substituting $R = r$ into equation (7). The coverage area of node z is approximately $\delta\ m^2$ assuming δ is small. The power required to receive $P_{c,r}$ is around the same as to transmit the data [11], so $P_{c,r}$ is equal to $P_{c,t}$. The number of nodes receiving a broadcast message from the cluster head is n since ϕ is approximately equal to n as described in Section III.B. As a

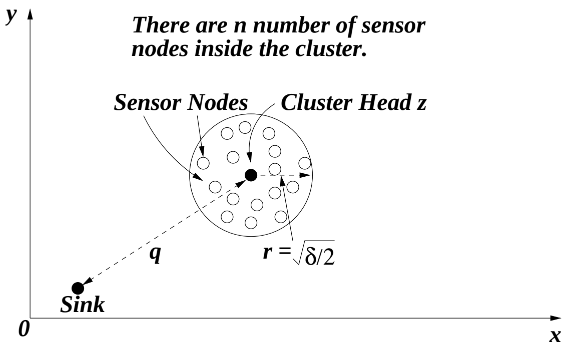


Fig. 17. Communication between sink and sensor nodes.

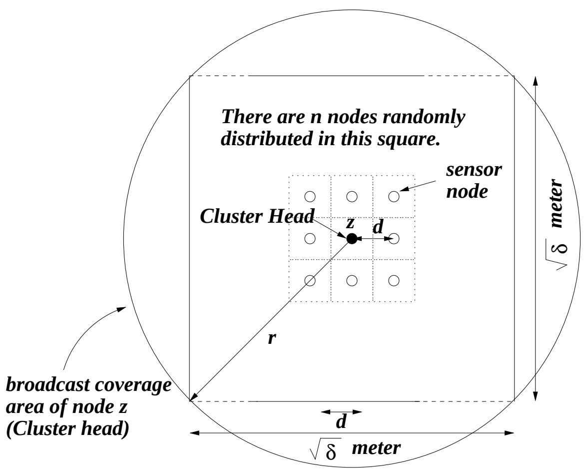


Fig. 18. Broadcast by sensor node in clustering.

result, the power required to distribute a task from the sink to the sensor nodes as shown in Figure 17 is calculated by equation (13).

$$P_{c,init} = P_{c,s} + n \cdot P_{c,r} + P_{c,t} + P_{c,process} \qquad (13)$$

where $P_{c,s}$ is the power used by the sink to transmit the task to the cluster head, and it is evaluated by substituting $\alpha = 1$, $R = q$, and $k = 2$ into equation (7); $P_{c,r}$ is the power required to receive the task, which is equal to $P_{c,t}$; $P_{c,t}$ is the power that the cluster head used to transmit the task to the sensor nodes; n is the number of nodes inside the cluster; and $P_{c,process}$ is the power used by all the nodes to route the task, which is assumed to be negligible since the amount of processing is small. After combining all the terms and assuming $P_{c,process}$ equals to 0, $P_{c,init}$ is calculated by equation (14).

$$P_{c,init} = P_r \left(\frac{4\pi}{\lambda}\right)^2 \cdot \left[\alpha(n+1)\left(\sqrt{\frac{\delta}{2}}\right)^k + q^2\right] \qquad (14)$$

We assume the task broadcast by the cluster head is only destined for one sensor node, and the transmission radius of the sensor node is r. As a result, the power P_{c,one_way} used to transmit data from the source to the sink according to Figure 17 is determined by equation (15).

$$P_{c,one\_way} = P_{c,t} + 2P_{c,r} + P_r \left(\frac{4\pi}{\lambda}\right)^2 q^2 \qquad (15)$$

where $P_{c,t}$ is the power used by the sensor node to transmit data to the cluster head; $P_{c,r}$ is the power needed to receive data from the sensor node or cluster head, which is assumed equal to $P_{c,t}$; and the last term on the right hand side is given by equation (7) with $\alpha = 1$, $R = q$, and $k = 2$, which is the power required by the cluster head to transmit data to the sink. Therefore, the total power consumed $P_{c,consume}$ in finding the sensor node to perform the task and periodically sending data from the source to the sink is calculated by equation (16).

$$P_{c,consume} = P_{c,init} + j \cdot P_{c,one\_way} \qquad (16)$$

where $P_{c,init}$ and P_{c,one_way} are given by equations (14) and (15), respectively ; and j is the number of times that the source sends data to the sink. After combining all the terms, $P_{c,consume}$ is calculated as follows:

$$\begin{aligned} P_{c,consume} &= P_r \left(\frac{4\pi}{\lambda}\right)^2 \left[\alpha(n+1+3j)\left(\sqrt{\frac{\delta}{2}}\right)^k \right. \\ &\quad \left. +(j+1)\left[(x_1-x_0)^2 + (y_1-y_0)^2\right]\right] \end{aligned} \qquad (17)$$

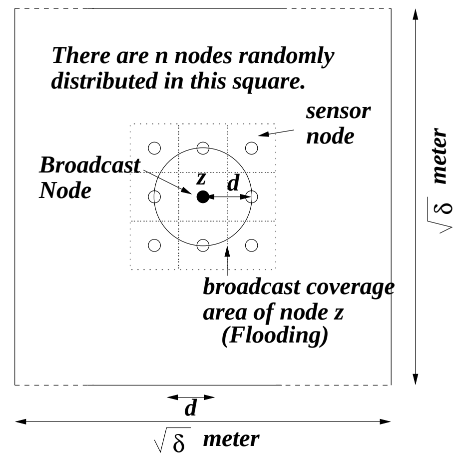


Fig. 19. Broadcast by a sensor node in flooding.

D. Power Consumption Based on SER

There are n nodes inside a $\sqrt{\delta}\ m$ by $\sqrt{\delta}\ m$ square as shown in Figure 19. The minimum broadcast distance of a sensor node z in Figure 19 is d, which is calculated by

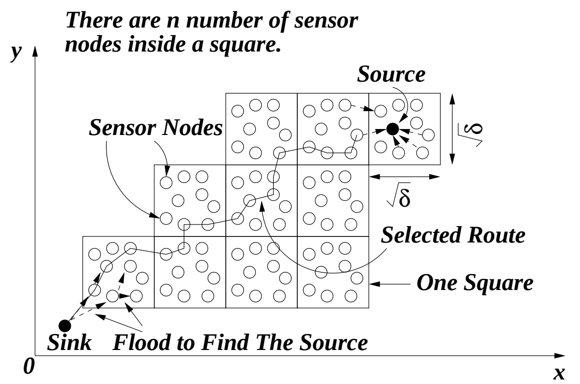


Fig. 20. Route setup and establishment with SER protocol.

equation (10). The sensor field consisting of M number of such squares is determined by equation (18).

$$M = \left\lceil \frac{A}{\delta\ m^2} \right\rceil \tag{18}$$

where A is evaluated by equation (8) and $\delta\ m^2$ is the area of the square.

Initially, the sink floods a task to all the sensor nodes in the sensor field as shown in Figure 20 to find the source. Each sensor node broadcasts the task only once regardless if its neighbors receive it or not. After the source is found, the source chooses the streams back to the sink according to the SER protocol as shown in Figure 20. Figure 20 shows the source only selects one stream at Level-1.

Since there are n nodes inside a $\sqrt{\delta}\ m$ by $\sqrt{\delta}\ m$ square, the total power used to transmit the task to all the nodes inside the square is $P_{f,t}$.

$$P_{f,t} = n \cdot P_t \tag{19}$$

where n is the number of nodes inside the square and P_t is calculated by equation (7). Each sensor node may receive the same task β times, depending on the broadcast distance. As a result, the total power used to receive the task is $P_{f,r}$.

$$P_{f,r} = \beta \cdot P_{f,t} \tag{20}$$

where β is the number of times that the node receives the same task and $P_{f,t}$ is determined by equation (19), because the power to receive is approximately equal to the power to transmit [11]. By combining $P_{f,t}$ and $P_{f,r}$ with the power used for processing the task $P_{f,process}$, which is negligible and assumed to be 0, the total power required to flood the task to the nodes in the $\sqrt{\delta}\ m$ by $\sqrt{\delta}\ m$ square is $P_{f,total}$.

$$P_{f,total} = P_{f,t} + P_{f,r} + P_{f,process} = P_{f,t}(\beta + 1) \tag{21}$$

where β is the number of times, which a sensor node receives the same task, and $P_{f,t}$ is calculated by equation (19). As a result, the total power required to flood the task to all nodes in the sensor field is:

$$P_{f,init} = M \cdot P_{f,total} \tag{22}$$

where M and $P_{f,total}$ are determined by equations (18) and (21).

After the source is found, a route is selected back to the sink as shown in Figure 20. The number of hops between the sink and the source is h_{total}. The power required to send data from the source to the sink is P_{f,one_way} when the power to broadcast and receive the data is the same in the sensor network environment [11].

$$P_{f,one\_way} = 2 \cdot h_{total} \cdot P_t \tag{23}$$

where h_{total} is the number of hops between the source and the sink, and P_t is given by equation (7). The total power consumed $P_{f,consume}$ in finding the targeted node and periodically sending data from the source to the sink is as follows:

$$P_{f,consume} = P_{f,init} + jP_{f,one\_way} + P_{f,select} \tag{24}$$

where $P_{f,init}$ and P_{f,one_way} are calculated by equations (22) and (23), respectively; j is the number of times that the source sends data to the sink; and $P_{f,select}$ is the power used to establish the selected route from the targeted node to the sink, and it is the same as P_{f,one_way}.

D.1 Minimum Power Consumption

The minimum value of $P_{f,consume}$ is obtained when the transmission radius R of a sensor node is equal to d, which is calculated by equation (10). The number of neighbor nodes that receive the signal from node z is 4 when the transmission radius is d as shown in Figure 19. As a result, β is equal to 4 in equation (20), and the number of hops h_{total} for all possible routes between the sink and the source is the same. The new h_{total} value is as follows:

$$h_{total} = \left\lceil \frac{|x_1 - x_0|}{d} \right\rceil + \left\lceil \frac{|y_1 - y_0|}{d} \right\rceil \tag{25}$$

where (x_0,y_0) and (x_1,y_1) are coordinates of the sink and source, respectively, and d is calculated by equation (10). The minimum $P_{f,consume}$ is obtained by substituting d, β and h_{total} into equations (7), (20) and (23), respectively, and rearranging the terms in equation (24).

$$P_{f,consume} = \alpha P_r \left(\frac{4\pi}{\lambda}\right)^2 \left(\sqrt{\frac{\delta}{n}}\right)^k [5Mn + 2(j+1)\left(\left\lceil \frac{|x_1 - x_0|}{\sqrt{\frac{\delta}{n}}} \right\rceil + \left\lceil \frac{|y_1 - y_0|}{\sqrt{\frac{\delta}{n}}} \right\rceil\right)] \quad (26)$$

E. Power Emission Level

The maximum power emitted by a sensor node while implementing the routing protocol based on the clustering techniques as described in Section III.C is $P_{cluster\_emission}$, which is determined by equation (2) with R, g_t, and g_r set equal to q, 1 and 1, respectively; q is the distance between the sink and source. $P_{cluster\_emission}$ is restated as follows:

$$P_{cluster\_emission} = P_r \left(\frac{4\pi}{\lambda}\right)^2 q^2 \quad (27)$$

where P_r is the required power at the receiver; λ is the wavelength of the transmitted signal; and q is the distance between the source and the sink calculated by equation (12).

One the other hand, the maximum power emitted by a sensor node $P_{SER\_emission}$ when the SER protocol is implemented is calculated by equation (7), and it is restated as follows:

$$P_{SER\_emission} = \alpha P_r \left(\frac{4\pi}{\lambda}\right)^2 R^k \quad (28)$$

where α is the additional attenuation constant as described in Section III.A; P_r is the power required at the receiver; λ is the wavelength of the transmitted signal; and R is the radius of transmission of a sensor node with a minimum value of d, which is calculated by equation (10).

IV. Performance Evaluation

A. By Analysis

We assume the sensor field is a 20 m by 20 m square, and δ as described in Section III.B is 1. Also, the sink is located at (0,0) and the source is located (20,20), so $x_1 - x_0$ and $y_1 - y_0$ are equal to 20. The area A of the sensor field is 400 m^2, and M as determined by equation (18) is 400.

A.1 Power Consumption Gain

The power consumption gain of the SER protocol versus the protocol based on clustering techniques as described in Section III.C is as follows:

$$G = \frac{P_{c,consume}}{P_{f,consume}} \quad (29)$$

where $P_{c,consume}$ and $P_{f,consume}$ are given by equations (17) and (24), respectively. The maximum power consumption gain G_{max} is obtained when the radius of transmission R is equal to d, which is calculated by equation (10). As a result, $P_{f,consume}$ in equation (29) is determined by equation (26).

$$G_{max} = \frac{\alpha(n+1+3j)\left(\sqrt{\frac{\delta}{2}}\right)^k + (j+1)\left[(x_1-x_0)^2 + (y_1-y_0)^2\right]}{\alpha\left(\sqrt{\frac{\delta}{n}}\right)^k \left[5Mn + 2(j+1)\left(\left\lceil \frac{|x_1-x_0|}{\sqrt{\frac{\delta}{n}}}\right\rceil + \left\lceil \frac{|y_1-y_0|}{\sqrt{\frac{\delta}{n}}}\right\rceil\right)\right]} \quad (30)$$

Substituting the values of x_0, x_1, y_0, y_1, M, and δ as given in Section IV.A, i.e., $x_0 = 0$, $x_1 = 20$, $y_0 = 0$, $y_1 = 20$, $M = 400$, and $\delta = 1$, into equation (30), G_{max} is determined by the following equation:

$$G_{max} = \frac{\alpha(n+1+3j)\left(\sqrt{\frac{1}{2}}\right)^k + 800(j+1)}{\alpha\left(\sqrt{\frac{1}{n}}\right)^k \left[2000n + 4(j+1)\left(\lceil 20\sqrt{n}\rceil\right)\right]} \quad (31)$$

where α is the additional attenuation constant in the sensor network environment; n is the density ξ, which is calculated by equation (9), inside a 1 m^2 square; j is the number of data transmission from the source to the sink; and k ranges from 2 to 4.

The maximum power consumption gain with α and j set equal to 2 and 20 while varying the value n is shown in Figure 21. G_{max} increases from around 8 dB as the density of nodes increases, and it increases more significantly for higher k value. From Figure 21, we know that the improvement is significant when the density of nodes ξ, which is calculated by equation (9), and the value of k are high.

Also, the number of times, which the source sends data to the sink, has an effect on the value of G_{max}. The value of G_{max} is positive when j is greater than 5 when α and n are set equal to 2 and 5, respectively, as shown in Figure 22.

A.2 Power Emission Gain

The power emission gain of SER $G_{emission}$ is calculated by equation (32).

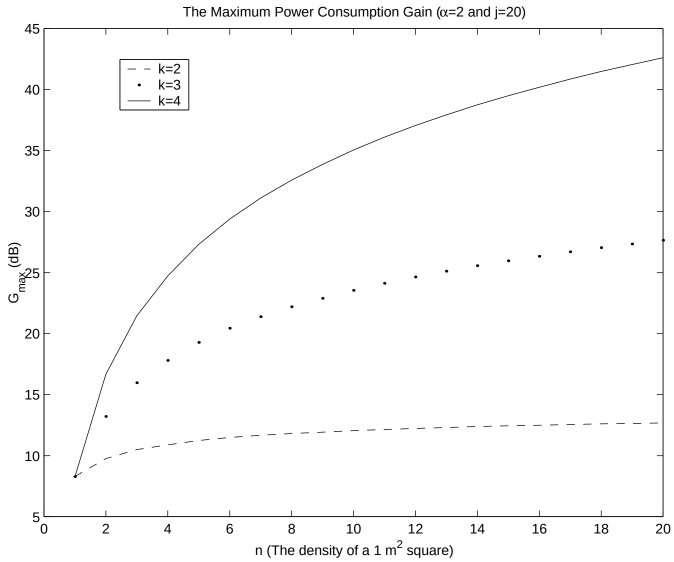


Fig. 21. G_{max} when $\alpha = 2$ and $j = 20$

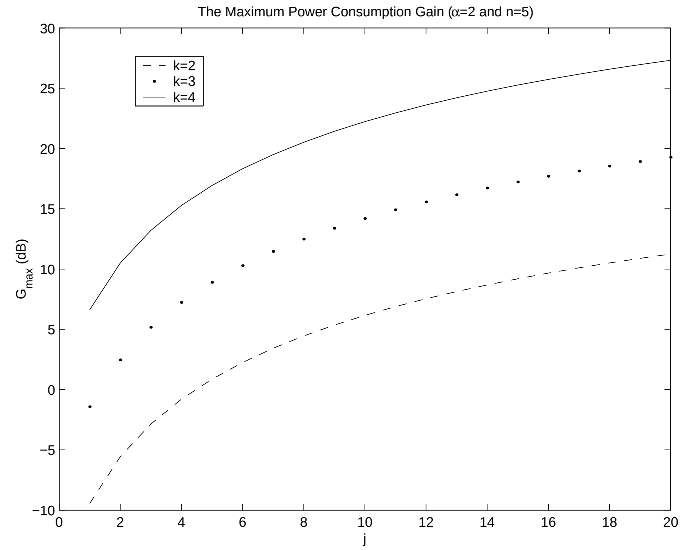


Fig. 22. G_{max} when $\alpha = 2$ and $n = 5$

$$G_{emission} = \frac{P_{cluster\_emission}}{P_{SER\_emission}} \quad (32)$$

where $P_{cluster\_emission}$ and $P_{SER\_emission}$ are calculated by equations (27) and (28). The maximum power emission gain is when the radius of transmission R is equal to d, which is calculated by equation (10). Substituting the values of x_0, x_1, y_0, y_1, and δ as given in Section IV.A into equation (32) and rearranging the terms, the maximum power emission gain $G_{max\_emission}$ is as follows:

$$G_{max\_emission} = \frac{800}{\alpha\left(\sqrt{\frac{1}{n}}\right)^k} \quad (33)$$

where α is the additional attenuation constant in the sensor network environment; and k ranges from 2 to 4.

B. By Simulation

The performance of the SER protocol is also evaluated with an event driven simulation. The performance data is collected from 50 simulation runs. One thousand non-mobile sensor nodes is deployed randomly in a 200 meters by 150 meters sensor field. Each of the sensor nodes can receive and transmit messages to its neighbors by executing the routing protocol independently, i.e., each sensor node is emulating a physical sensor node where it has its own memory and routing state. When a node receives and transmits messages, it will consume power. It does not consume power when it is idle, i.e., when there is no message to receive or transmit. The sink and source nodes are located at (0,0) and (180,130) of the sensor field. The configuration of each node is listed in Table II.

Parameters	Value
Transmission radius	10 meters
Available energy	1 Joule
Transmission cost	600 mW
Receiving cost	200 mW
Transmission frequency	2 MHz
Transmission bandwidth	1 MHz
Signal propagation speed	$3 * 10^8$ meters/second
Time required to process outgoing message	0.02 seconds
Time required to process incoming message	0.01 seconds

TABLE II
CONFIGURATION OF EACH SENSOR NODE.

The SER protocol is compared to the flooding, gossiping, and SPIN1 [5] protocols in Section IV.B.1. The flooding protocol does not require a node to have a unique ID in order to identify the neighbors of the node, i.e., the maximum number of IDs assigned to sensor nodes is equaled to the number of nodes deployed. On the other hand, the gossiping and SPIN1 protocols do require a unique ID, because both of them need to know the exact neighbor that the message is intended. As for the SER protocol, it only uses 800 IDs when deploying 1000 nodes in all the simulation runs. A more in-depth analysis of the SER protocol is discussed in Section IV.B.2, e.g., the effect of the sleep mode operation being turned ON and the number of sensor nodes deployed being increased while the ID range remains at 800.

The following is a table listing the length of each message used in different protocols.

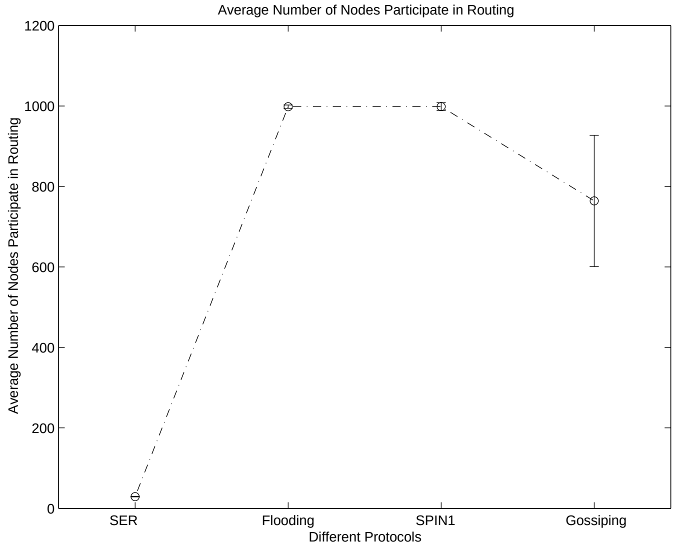


Fig. 23. Number of nodes participate in routing for different protocols.

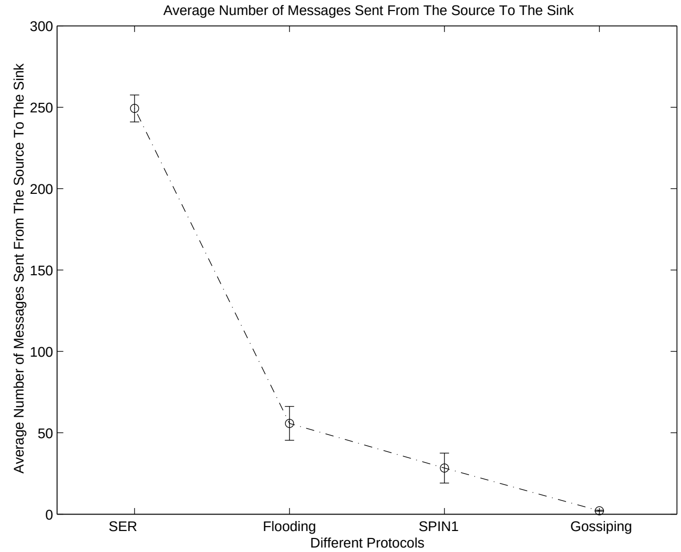


Fig. 24. The number of messages sent in different protocols.

Protocols	Message And Its Length In Bits
SER	*N-message*, *S-message*, *U-message*, and *I-message* are 4000 bits.
Flooding	The data message is 4000 bits.
SPIN1	The ADV and REQ messages are 128 bits; the DATA message is 4000 bits.
Gossiping	The data message is 4000 bits.

TABLE III
LIST OF MESSAGES AND ITS LENGTH USED IN DIFFERENT PROTOCOLS.

B.1 Comparison of different protocols

The number of sensor nodes participated in routing messages from the source node to the sink node is close to the number of nodes deployed, i.e., 1000, when flooding and SPIN1 protocols are used as shown in Figure 23. As for the gossiping protocol, it should reach the 1000 level; it does reach that level, because the gossiping protocol takes long time to disseminate the message to all nodes. As a result, the simulations have to be ended early. The large standard deviation from the average as shown in Figure 23 validates this situation. While flooding, SPIN1, and gossiping protocols involve around 1000 nodes to send a message from the sink to the source, the SER protocol only requires around 30 sensor nodes when stream $S(1,1)_{1,0}$ is used.

Since flooding, SPIN1, and gossiping protocols use data dissemination approach to send data from the source to the sink, the energy of the network is depleted faster than when SER protocol is used. To validate this, a message is sent from the source to the sink every 10 seconds. If the sensor nodes use the SER protocol to route the messages, an average of 249 messages as shown in Figure 24 reaches the sink successfully before the route is broken. If the route is broken and the sink wants to get more data from the source, the sink can initiate another route setup by broadcasting a *S-message*. On the other hand, the average number of messages successfully received for flooding, SPIN1 and gossiping protocols are around 56, 28, and 2, respectively. The reason for the low performance of SPIN1 protocol as compared to flooding protocol is because SPIN1 protocol uses a handshake of ADV, REQ, and DATA messages in a wireless network, where the node density is high, and nodes that are not interested in the broadcast overhear the handshake messages. As shown in Figure 25, the SER protocol consumes the least amount of network energy per message, and the gossiping protocol consumes the most with large standard deviation from the mean. The network energy consumed per message with respect to time is plotted in Figure 26. The performance of flooding and SPIN1 protocols are comparable while around 7 percent of the network energy is consumed when gossiping protocol is used. The performance of the SER protocol is the best one out of the four.

One other important characteristic of a routing protocol is the time required for a message to reach the sink from the source. The performance of this characteristic is illustrated in Figures 27 and 28. The gossiping protocol takes a long time to reach the sink; it takes around 70 seconds and has large standard deviation, i.e., jitter. A message takes the shortest time when flooding protocol is used, but the jitter is the highest among SER, flooding, and SPIN1 pro-

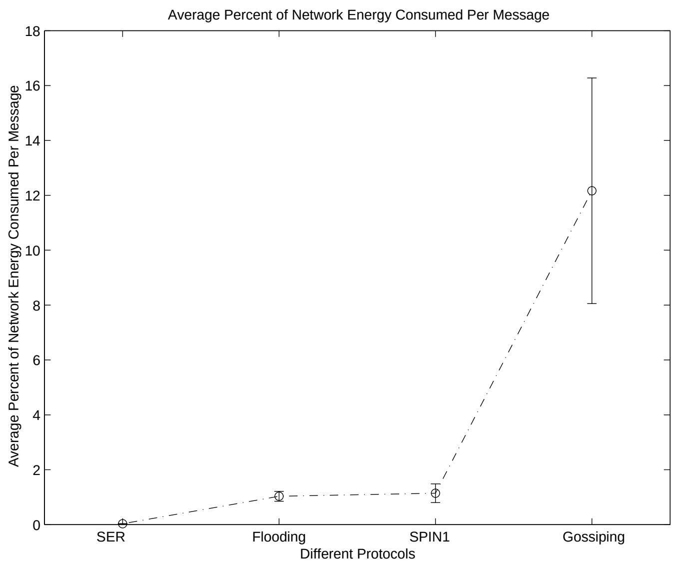


Fig. 25. The network energy consumed when different protocols are used.

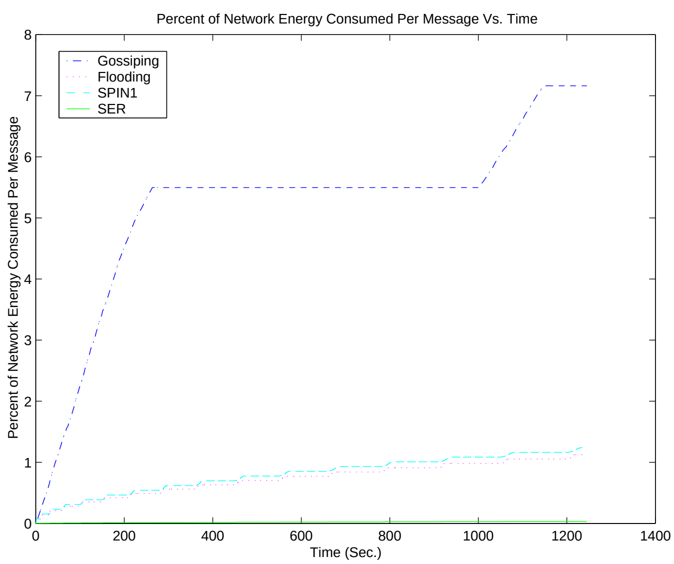


Fig. 26. The network energy consumed with respect to time.

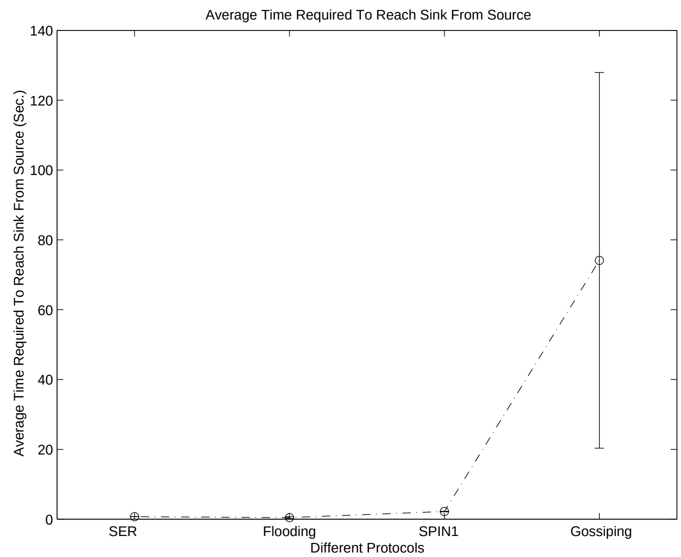


Fig. 27. The time required to reach the sink from the source.

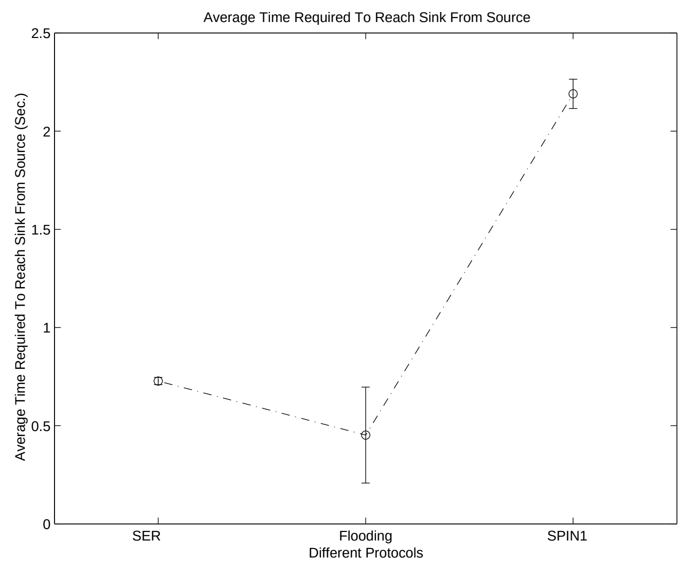


Fig. 28. The time required to reach the sink for SER, flooding, and SPIN1 protocols.

tocols. The SER protocol takes around 0.73 seconds and has the smallest jitter, i.e., 0.02 seconds, while the flooding protocol needs 0.45 seconds with 0.24 seconds of jitter.

B.2 In-depth performance evaluation of the SER protocol

The in-depth performance evaluation is separated into three parts; the first part evaluates the performance of the SER protocol when the sleep mode operation is turned OFF, and the second part evaluates when the sleep mode operation is turned ON; lastly, the third part evaluates the SER protocol when the number of nodes deployed is increased.

B.2.a Sleep Mode Operation OFF

. The SER protocol allows the source to choose the type and level of the streams to carry the messages to the sink. As shown in Figure 29, the type 2 and 4 streams involve more nodes than type 1 and 3 streams regardless if they are at Level-1 or Level-2. It is because type 2 and 4 streams require more than one stream to route the messages. From Figure 29, the data also indicates that the streams merged into one stream at some point between the source and the sink. As the stream width is increased to 1 or 2, i.e., at Level-2, the number of nodes involved in the stream increases. By increasing the width of the streams or choosing multiple streams, i.e., type 2 or 4, to route the messages , the streams are more robust to sensor node failure, but

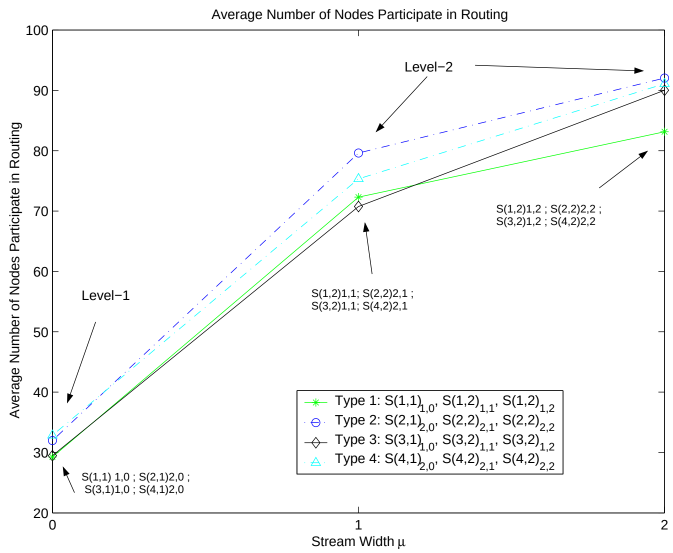


Fig. 29. The number of nodes required for each stream.

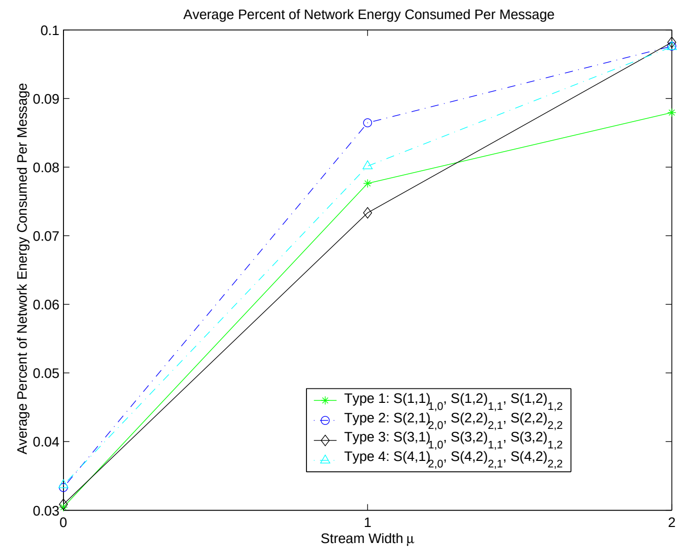


Fig. 31. The network energy consumption for each stream.

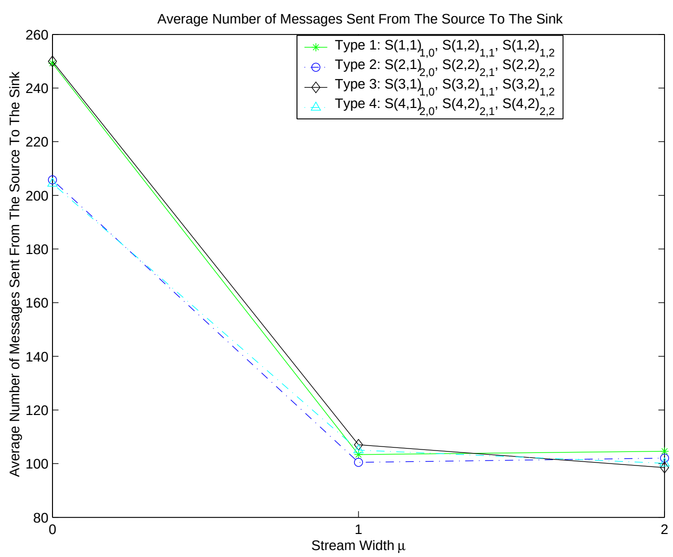


Fig. 30. The number of message sent for each stream.

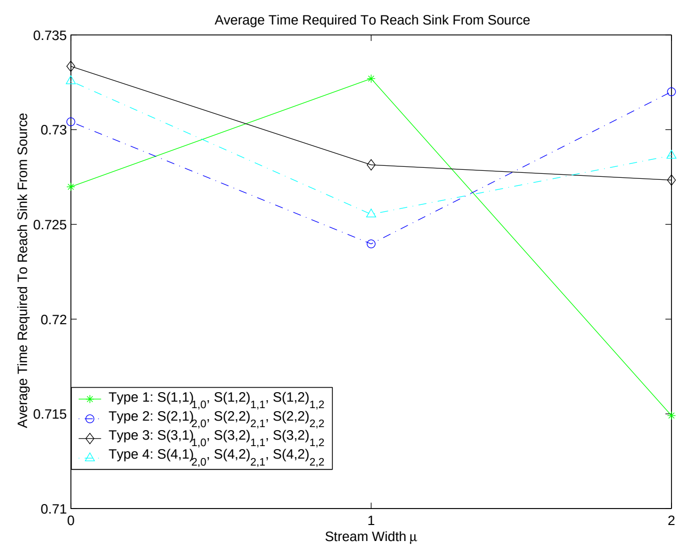


Fig. 32. The time required to reach the sink for each stream.

the average number of messages that can be sent with the streams decreases as shown in Figure 30. The results also indicate that the stream width can increase to 2 without decreasing the ability to send messages while increasing robustness. With increased robustness, there is a tradeoff, which is the network energy consumption, as given in Figure 31.

As shown in Figures 32 and 33, the $S(1,1)_{1,0}$ stream does provide the shortest time to reach the sink with the smallest jitter when the stream width is 0, i.e., at Level-1. The $S(1,1)_{1,0}$ stream is intended to carry time sensitive messages. As the width of the stream increases, the time required to reach the sink for all the streams seems not predictable, but it is bounded within 0.735 seconds and the jitter is within 0.065 seconds. Note that the main purpose of a Level-2 stream, i.e., $\mu > 0$, is to increase the robustness of the stream and not to optimize the time-of-arrival.

B.2.b Sleep Mode Operation ON

. If all the sensor nodes turn ON the sleep mode operation, i.e., a sensor node turns OFF the receiver for 1 seconds after it receives an *I-message*, the number of message that can be sent through the streams increases by 26 to 82 percent. The number of messages sent from the source to the sink with sleep mode operation OFF and ON is shown in Figures 30 and 34. The network energy consumption per message is also lower when the sleep mode operation is ON. The figures, which show this difference, are given in Figures 31 and 35 for sleep mode operation OFF and ON, respectively. As for the time required to reach the sink, the

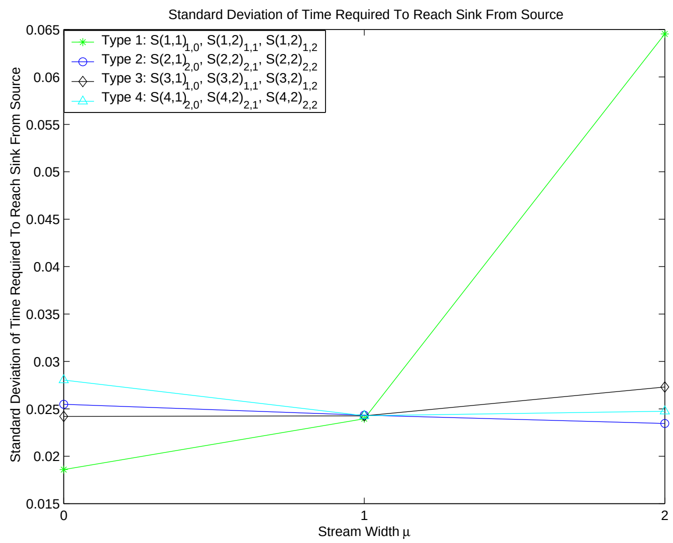


Fig. 33. The jitter of the time required to reach the sink.

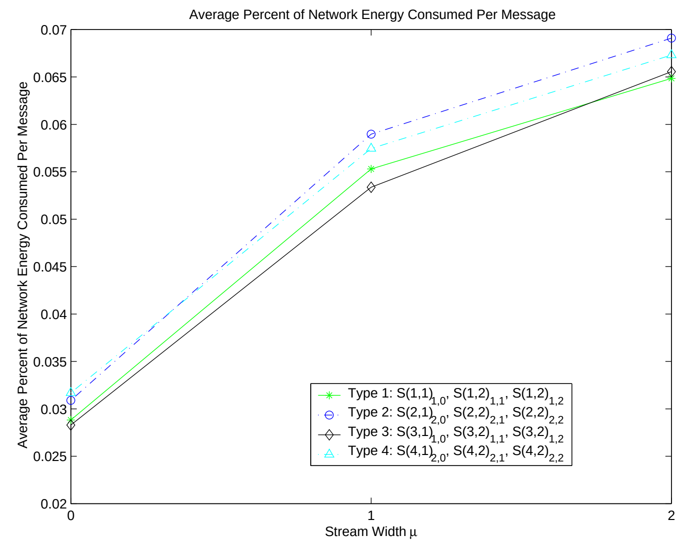


Fig. 35. The network energy consumption when the sleep mode operation is ON.

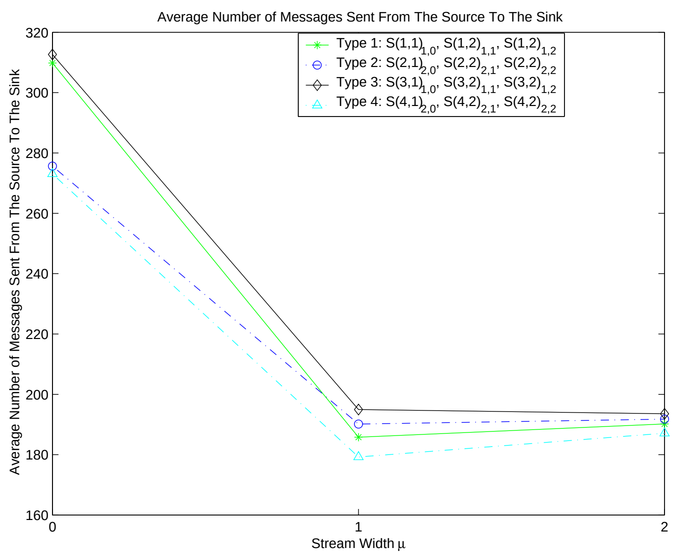


Fig. 34. The number of messages sent when the sleep mode operation is ON.

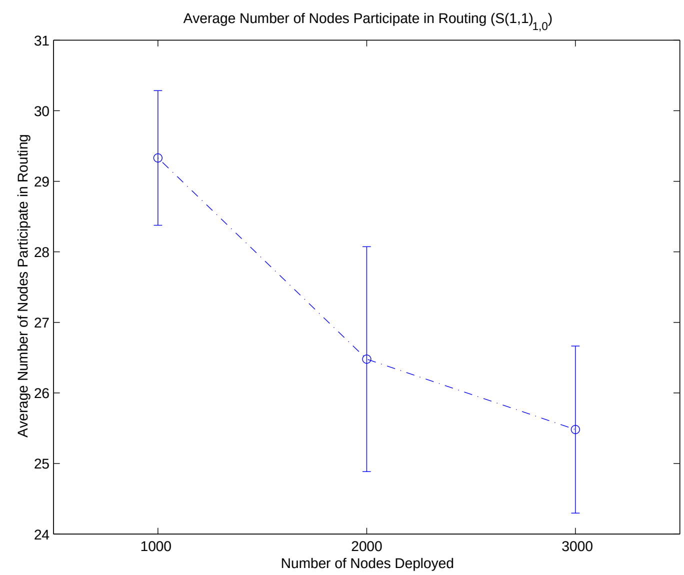


Fig. 36. The number of nodes participate in routing as the number of nodes deployed increases.

characteristic remains the same but the jitter is bounded within 0.029 seconds as compared to 0.065 seconds.

B.2.c Increased Number of Nodes Deployed

. Simulations are also performed to test the SER protocol when the number of nodes being deployed is increased. The $S(1,1)_{1,0}$ stream, which is a type 1 and Level-1 stream intended to route time sensitive messages, is used for such analysis. For all the simulation, the sleep mode operation is also turned ON. As the number of nodes increases, the average number of sensor nodes participating in the routing decreases as given in Figure 36. This indicates that the SER protocol is creating a stream along a straight line between the source and the sink; the stream consists of sensor nodes that are near the edge of the broadcast radius, i.e., 10 meters. The source is located at (180,130), and the ideal minimum number of nodes participating in the stream along the straight line is 22. From Figure 36, the average number of nodes is approaching this ideal value as the number of nodes deployed increases. By having more sensor nodes in the sensor field, the average number of message, which can be sent, is not affected as much but with only a slight decrease as shown in Figure 37.

With the decrease in the number of nodes participating in the stream, the average percent of network energy consumed per message also decreases as illustrated in Figure 38. The average time required to reach the sink from the

Fig. 37. The number of messages sent as the number of nodes deployed increases.

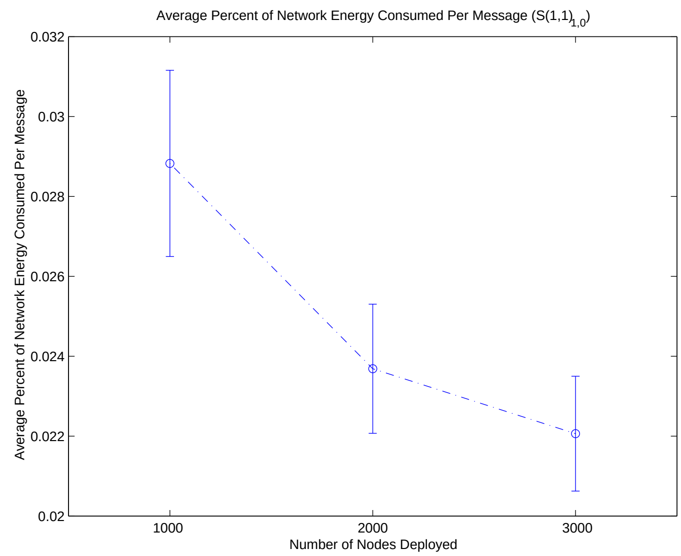


Fig. 38. The network energy consumption as the number of nodes deployed increases.

source also decreases as the number of nodes participating in the stream approaches the ideal value, i.e., 22, as shown in Figure 39. Since the straight line between the source and the sink is the shortest path, the result does match the expectation that the shortest path takes the least amount of time to route messages. Note that the jitter also decreases when the number of nodes deployed increases. By increasing the number of nodes in the sensor field, the time required to reach the sink as well as the variation of this time decreases. Also, the range of IDs used in all the simulation still remains at 800 as the number of nodes deployed increases to 2000 and 3000. This shows another important aspect of sensor networks; sensor nodes should use local IDs instead of unique global ID to conserve energy as well as memory. The simulation results show that the SER protocol embraces this local ID requirement and allows the density of the sensor network to be scalable without affecting the functionality of the protocol.

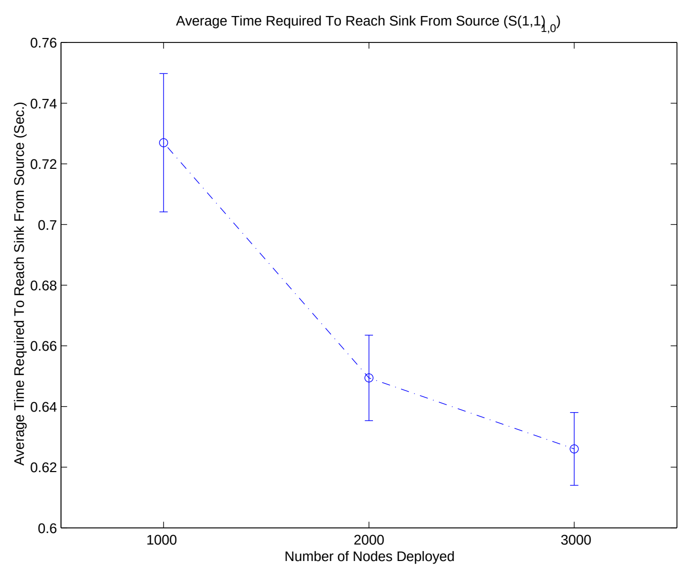


Fig. 39. The time required to reach the sink as the number of nodes deployed increases.

V. Conclusion

We introduced a new routing protocol called SER. In this protocol, the sink floods the task to the sensor nodes in the sensor field to find the sources. After the sources are found, they select the routes back to the sink. We showed that SER is more power effective than a protocol based on clustering techniques. The SER shows a maximum power consumption gain of 43 dB as given in Figure 21. Also, the maximum power consumption gain G_{max} in dB is positive when the sources need to send more than 5 messages to the sink for $\alpha = 2$ and $\xi = n = 5$ as shown in Figure 22. We also showed that the maximum power emission level is much less in Section IV.A.2. For $\alpha = 2$, $\xi = n = 5$, and $k = 3$, the maximum power emission gain $G_{max\_emission}$ as calculated by equation (33) in dB is 84 dB.

We also verified by simulations that the SER protocol is more energy efficient than flooding, SPIN1, and gossiping protocols. In addition, the average time required to reach the sink is the second lowest among the 4 protocols, but the SER protocol has the smallest amount of jitter. Also, when the sleep mode operation is turned ON, the number of message that can be sent through the streams increases by 26 to 82 percent. Using the stream that is designed to carry time sensitive messages, i.e., $S(1,1)_{1,0}$, the number of nodes participating in routing, the time required to reach the sink, and the jitter of the time-of-arrival decrease as more nodes are deployed in the sensor field. In addition, the SER protocol does not require each node to have an unique ID. As a result, only a small range of IDs is needed regardless if the number of sensor nodes is increased.

REFERENCES

[1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E., ”Wireless Sensor Networks: A Survey,” *Computer Networks (Elsevier) Journal,* pp. 393-422, March 2002.

[2] N. Bulusu, D. Estrin, L. Girod, and J. Heidemann, ”Scalable Coordination of Wireless Sensor Networks: Self-Configuring Localization Systems,” *International Symposium on Communication Theory and Applications (ISCTA 2001),* Ambleside, UK, July 2001.

[3] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, ”Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks,” *Proc. of the ACM MobiCom’01,* pp. 85-96, Rome, Italy, 2001.

[4] S. Hedetniemi, S. Hedetniemi, and A. Liestman, ”A Survey of Gossiping and Broadcasting in Communication Networks,” *Networks,* 18, 1988.

[5] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive Protocols for Information Dissemination in Wireless Sensor Networks,” *Proc. of the ACM MobiCom’99,* pp. 174-185, Seattle, Washington, 1999.

[6] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, ”Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” *IEEE Proceedings of the Hawaii International Conference on System Sciences,* pp. 1-10, January 2000.

[7] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks,” *Proc. of the ACM MobiCom’00,* pp. 56-57, Boston, MA, 2000.

[8] L. J. Ippolito Jr., ”Radiowave Propagation in Satellite Communications,” *Van Nostrand Reinhold Company,* 1986.

[9] J. M. Kahn, R. H. Katz, and K. S. J. Pister, “Next Century Challenges: Mobile Networking for “Smart Dust”,” *Proc. of the ACM MobiCom’99,* pp. 271-278, Seattle, Washington, 1999.

[10] C. Perkins, ”Ad Hoc Networks,” *Addison-Wesley,* 2000.

[11] G.J. Pottie and W.J. Kaiser, “Wireless Integrated Network Sensors,” *Communications of the ACM,* vol. 43, no. 5, pp. 551-8, May 2000.

[12] A. Savvides, C. Han, and M. Srivastava, “Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors,” *Proc. of the ACM MobiCom’01,* pp. 166-179, Rome, Italy, 2001.

[13] S. Singh, M. Woo, and C. S. Raghavendra, “Power-Aware Routing in Mobile Ad Hoc Networks,” *Proc. of the ACM MobiCom’98,* pp. 181-190, Dallas, Texas, 1998.

[14] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, ”Protocols for Self-Organization of a Wireless Sensor Network,” *IEEE Personal Communications,* pp. 16-27, October 2000.

[15] Y. Xu, J. Heidemann, D. Estrin, ”Geography-informed energy conservation for ad hoc routing,” *Proc. of the ACM MobiCom’01,* pp. 70-84, Rome, Italy, 2001.