Egyptian Informatics Journal (2012) **13**, 155–169

Cairo University

Egyptian Informatics Journal

www.elsevier.com/locate/eij
www.sciencedirect.com

ORIGINAL ARTICLE

Designing a local path repair algorithm for directed diffusion protocol

Basma M. Mohammad El-Basioni *, Sherine M. Abd El-kader, Hussein S. Eissa

Computers and Systems Department, Electronics Research Institute, Cairo, Egypt

Received 6 April 2012; revised 11 July 2012; accepted 16 July 2012
Available online 11 August 2012

KEYWORDS

Attribute-value based naming scheme;
Data-centric routing;
Data gathering;
Energy-efficiency;
Locality;
Wireless sensor network

Abstract This paper proposes an implementation for the directed diffusion paradigm aids in studying this paradigm's operations and evaluates its behavior according to this implementation. The directed diffusion is evaluated with respect to the loss percentage, lifetime, end-to-end delay, and throughput. From these evaluations some suggestions and modifications are proposed to improve the directed diffusion behavior according to this implementation with respect to these metrics. The proposed modifications reflect the effect of local path repair by introducing a technique called Loop-free Local Path Repair (LLPR) which improves the directed diffusion behavior especially with respect to packet loss percentage by about 92.69%. Also LLPR improves the throughput and end-to-end delay by about 55.31% and 14.06% respectively, while the lifetime decreases by about 29.79%.

1. Introduction

Directed diffusion [1–6] is a data gathering and dissemination paradigm for Wireless Sensor Networks (WSNs) [7–10]. Directed diffusion is characterized by data-centric routing, application-aware processing, attribute-value based naming scheme, publish-and-subscribe information model, intermediate nodes might aggregate data, locality, robustness, and energy-efficiency. Directed diffusion uses a publish-and-subscribe information model in which an inquirer expresses an interest using attribute–value pairs. Sensor nodes, which can service the interest, reply with the corresponding data. The main elements of direct diffusion include interests, data messages, gradients, and reinforcements.

An interest can be viewed as a query or an interrogation that specifies what the inquirer wants. A gradient can be thought of as a reply link pointing toward the neighboring node from which the interest is received. The data message is the event description. Path reinforcement process is the process in which the sink can use multiple paths it established during the gradient setup phase to higher quality events by increasing its data rate [11].

* Corresponding author. Address: Computers and Systems Department, Electronics Research Institute, 33 El-Tahrir St., El-Dokki, Giza 12622, Egypt. Tel.: +20 10 303 557 3; fax: +20 2 333 516 31.
E-mail addresses: bbasioni@yahoo.com, bbasioni@eri.sci.eg (B.M. Mohammad El-Basioni).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.eij.2012.07.001

The design space of directed diffusion is wide, fundamentally, directed diffusion is a general pattern for communications in WSN. It specifies the headlines for sensor nodes to communicate and achieve certain goals from its communication, but the details of this pattern, how these headlines are implemented, and how the resulted trade-offs among design goals are explored tolerate a lot of possibilities.

This paper implements by simulation an instantiation from directed diffusion for tracking applications, so that, more than one sink can exist in the network in any place in the sensor field, each one can send a different request for data, and the sensor nodes can sense more than one object from the same or different type of data. This instantiation is evaluated with respect to the loss percentage, lifetime, end-to-end delay, and throughput. Some modifications to this implementation are suggested to improve its performance and the effects of these modifications on the paradigm are studied.

The rest of the paper is organized as follows; Section 2 describes the directed diffusion instantiation implemented in this paper, Section 3 mentions some contemplations and recommendations for directed diffusion implementation, Section 4 represents the loop-free local path repair implementation, Section 5 represents critical assessment of LLPR, Section 6, evaluates this instantiation, and finally Section 7 represents the conclusions and directions for future work.

2. Description of directed diffusion instantiation

This section describes the implementation of directed diffusion used in this paper. First, it recites the types of messages transmitted during network operation and the tables needed by sinks and nodes to store the information required by the paradigm, and then it demonstrates the operation of directed diffusion.

2.1. The messages sent

The original interest message (Interest_Msg): represents the sink request for data.

The data message (Data_Msg): contains the source's sensed data and includes fields for timestamp, count of sensed targets, a specific value for each target of a certain property characterizes targets of this data type, and the distances of the originator source from each target.

The positive reinforcement message (Reinf_Interest_Msg): is the same as the Interest_Msg but with the reinforced values for the Reporting Interval (RI) and Expiration time (Exp. time) and with fields for the IDs of the reinforced neighbor and the source reinforced by this message.

The negative reinforcement message (Neg_Reinf_Interest_Msg): is the same as the Reinf_Interest_Msg but with the negative reinforced values for the RI and Exp. time.

2.2. Tables held by nodes

The interest cache (Interest_Cache): contains information about the data requests received at the node and it includes records for each interest include, the count of Gradients for this data request and Gradients' information, the count of data sources replied with the requested data through each Gradient, and related information such as a flag to indicate whether this source is positively or negatively reinforced with respect to the IR and/or Exp. time through this Gradient for this data interest or not and the IDs of the sinks positively reinforced it if it is positively reinforced.

The data cache (Data_Cache): it stores sources' data information and it includes records for each Data_Msg include an array (AllConveyNeighs) for the information of all the neighbors conveyed each Data_Msg in the time order they conveyed it; an awareness should be given for conserving this array contents of the first data message conveyed from each source if the Data_Cache is refreshed.

The positively reinforced sources' table (Reinforced_Sources): the Reinforced_Sources table of a node saves the IDs of the data sources positively reinforced before through this node and the current and old index of the reinforced neighbor for each one of these sources in the AllConveyNeighs array of the first data message conveyed from each source.

2.3. Tables held by sinks

(Interests_Sent) table: this table stores the information of the Interest_Msgs which the sink sent, the source-neighbor pairs records which represent the sources replied to each interest and the sink's neighbors conveyed the first Data_Msg reply of each source, and other related information.

(Sources_Replied) table: this table stores the information of the data sources sent data to the sink including currently sensed objects of different data types and their corresponding distances from the source and the time of the latest reinforcement message the sink sent to it for path repair.

(Sinistral_Sources) table: this table contains the information of the sinistral sources which did not send any data to the sink for a time period greater than or equal to a specified period; this table includes fields for the ID, a flag to indicate if lastly a positive reinforcement was sent to this source for path repair, and if one sent, to which neighbor it was sent, flags to indicate if the source sent data in its last negative reinforcement interval, and if it sent, if the same reinforced neighbor for this source conveyed the source negative data. The first three fields are updated in each time of sinistral sources check, while the last two flags are updated during each interval of the negative data sending for each source.

(Sensed_Objects) tables: these are a set of tables, one for each object the sink was informed about it identified by its data type and an ID mapped to its unique value of the property used to characterize the same data type objects. Each table contains information related to each object.

The sink may also store the information it received in different formats and in different tabular structures such that its representation becomes more useful and easier to be retrieved by the used applications whatever these applications are.

2.4. The operation of directed diffusion instantiation

The sink always does the check for cut paths (Sink Path Repair (SPR)) when it receives a Data_Msg; the sink does the check for cut paths to see if there is a problem in a path to a source, if this problem is a cut in the path so a path repair is required, or it is just a temporary breakdown in the path, so a path repair is not required. It does that by looping through the Sources_Replied table.

If the sink found the difference between the incoming message timestamp and the timestamp of the latest Data_Msg of a recorded source is greater than a specified period measured in terms of the positively reinforced value for the RI and the time passed from the last Reinf_Interest_Msg sent to this source is also greater than a specified period, if this correct, it understands that the path is cut, so it adds this source to the Sinistral_Sources table.

But also it does not send a Reinf_Interest_Msg for path repair until it found that after the source no longer sends data, at least one of its sensed targets is sensed by less than three sources to be sure that its reading is important for accurate specification for at least one target position. Also, the sink considers the probability that there is no longer data from this source to send about the only object it senses as it went out of its sensing range, so it sends the reinforcement message if the count of targets lastly sensed by the source is greater than one or equals one and this source is the only one senses this object from the object appearance time, or it is not the only source but the last sources sensed this object sensed it from a specified period its pass indicates with a great probability that this object moved out the requested area or becomes inexistent for any reason.

Also in this condition, if the sink found that the latest source still sends data but without this object data or it sends no data from that considerable time and the count of objects it sensed at that time equals one and the count of sources sense this object in this time is greater than three, it will not send the repair message. If it found this last source sensed this object from a considerable time, did not send any data from that time and the count of objects it sensed at that time is greater than one, it will send the repair message to the neighbor which should be reinforced.

In case of the last sources sensed this object sensed it not from a large period and the sensing circles corresponding to the sensing distances of these sources and the sensing circle corresponding to the maximum sensing range of any node and its center is the sinistral source are separate, not coincident, and not contained in each other, the sink will not send the message.

If the sink sent a message, it marks this neighbor as positively reinforced for this source for all requests, and sets the time of last Reinf_Interest_Msg sent to this source to the current time, also it updates the Sinistral_Sources table to indicate that a Reinf_Interest_Msg for path repair was sent for this source and to which neighbor.

The decision of sending or not sending the Reinf_Interest_Msg for path repair at this time may be incorrect, but it is necessary to be taken at this time and it should be taken after a test to increase the probability of the decision correctness and accordingly decrease the probability of data loss of an important source, energy loss, and traffic increase for performing path repair for a path became useless. So, the sending of data at negatively reinforced intervals is exploited to know the extent of the correctness of this decision by updating the Sinistral_Sources table during the negatively reinforced intervals for each source and making a test at the end of this interval.

The test is performed such that, if the sinistral source did not send negative data through this interval, it is deleted from the Sources_Replied table; if it sent and a reinforcement for repair was sent to it already, the sink checks if the same reinforced neighbor sent, it sets the neighbor to be reinforced if this source makes another problem to the same neighbor, if not, it increments the index of the neighbor to be reinforced, so that the next neighbor in the source-neighbor pairs records is sent the reinforcement message if this source makes another problem. If this source sent and the taken decision was not to send reinforcement for repair and from this test it was found that this decision is wrong, the sink will send the reinforcement immediately to the appropriate neighbor and update the tables should be updated.

The flow charts depicted in Figures 1 and 2, describe parts of nodes' operation with respect to Data_Msg processing and Reinf_Interest_Msg processing in directed diffusion instantiation respectively.

3. Contemplations, notes, comments, and recommendations for directed diffusion implementation

- It is not necessary for the selection of the sink and accordingly the selection of every node in the reinforced path to the first neighbor conveyed reply from a specific source to be positively reinforced, to result in the selection of the low delay path because the end-to-end delay does not depend only on the transmission delay and the propagation delay for the data packet through each hop, but it also depends on the queuing delay which will be changed when there will be a single path to the sink for this source. Also, when there is a reinforced path specific to each source for each sink, one or more of these reinforced paths may be criss-cross with others or a part or a whole of this path coincides on others, then the queuing delay will increase the delay of these paths upon which they were selected.
- The sink should delay the start of negative reinforcement messages transmission for a certain delay, may be set to be different and suitable for different initial nodal densities, to ensure that the flooding of the first data message is completely ended, so, each node received it from all of its neighbors convey this source first data message to it, thus when a node receives the negative reinforcement, it will send it to all of them. Also, the sink delays the start of negative reinforcement messages transmission to ensure that the positive reinforcement for this source and other sources their first data received at the sink in the same time period sent and reached or is impending to reach them.
- The start of the negative reinforcement reporting interval for specific source data may be different for the Gradients requested this data and this source sent it through them before. So, the sample of a source transmitted through multipath may be a sample with different timestamp than that transmitted through multipath of other sources even though they sent the initial sample with each other with the same timestamp; the source itself may perform its multipath transmission with different samples with different timestamps, accordingly, the preparation period for the taken path repair decision correctness test for each source should be set larger when the node density is larger, as the variation in the start of the negative reinforcement reporting interval for different Gradients and different sources increases.
- In the initial data propagation, if each node resends the same data message it received from its neighbors to the Gradients requested this data, this will represent wasted

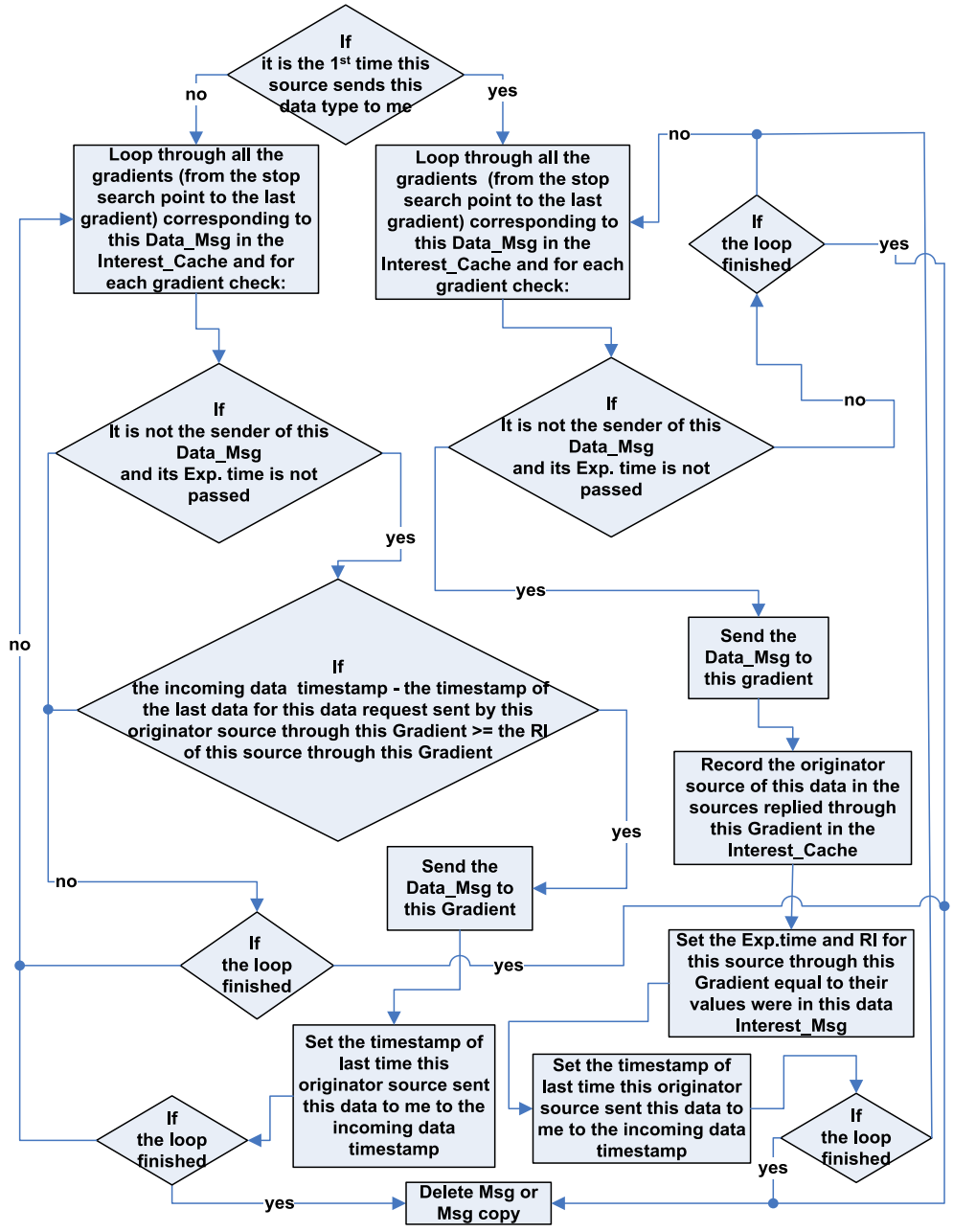

Figure 1 A part of Data_Msg processing by nodes in directed diffusion instantiation.

energy from the network, require large storage capacity from a node, and result in delay, so, each node resends the data message only once when it is received from the first neighbor conveyed it, but it records the IDs of all the neighbors conveyed this data message to it for the purpose of negative and positive reinforcements and path repair.

- This method of benefiting from the negative reinforcement in checking the extent of correctness of the path repair decision taken lastly for a certain source may delay the sink in detecting the death of the sources and the neighbors that positively reinforced for them or in detecting the cut of all the paths lead to a source from the sink or from the neighbor already reinforced for this source to it, especially when one of these conditions such as source death done immediately after it sends its negative reinforced data sample. So the negative reinforcement RI should not be short to not burden the network in sending a lot of redundant data, and in the same time it should not be long to fasten the repair decision correction and fasten the reception of sources data at the sink if it did not receive any data from them from a considerable period of time.
- In the period before a source receives any positive reinforcement, a data sample with smaller timestamp of a certain source may be received at the sink lately and after the reception of a data sample with larger timestamp of another source, this causes the sink to wrongly consider this late source as a sinistral source and it initiates a path repair which wastes the usages of some healthy paths and as a result reduces the count of probable paths to sources, which may reduce even slightly the network lifetime. So it is

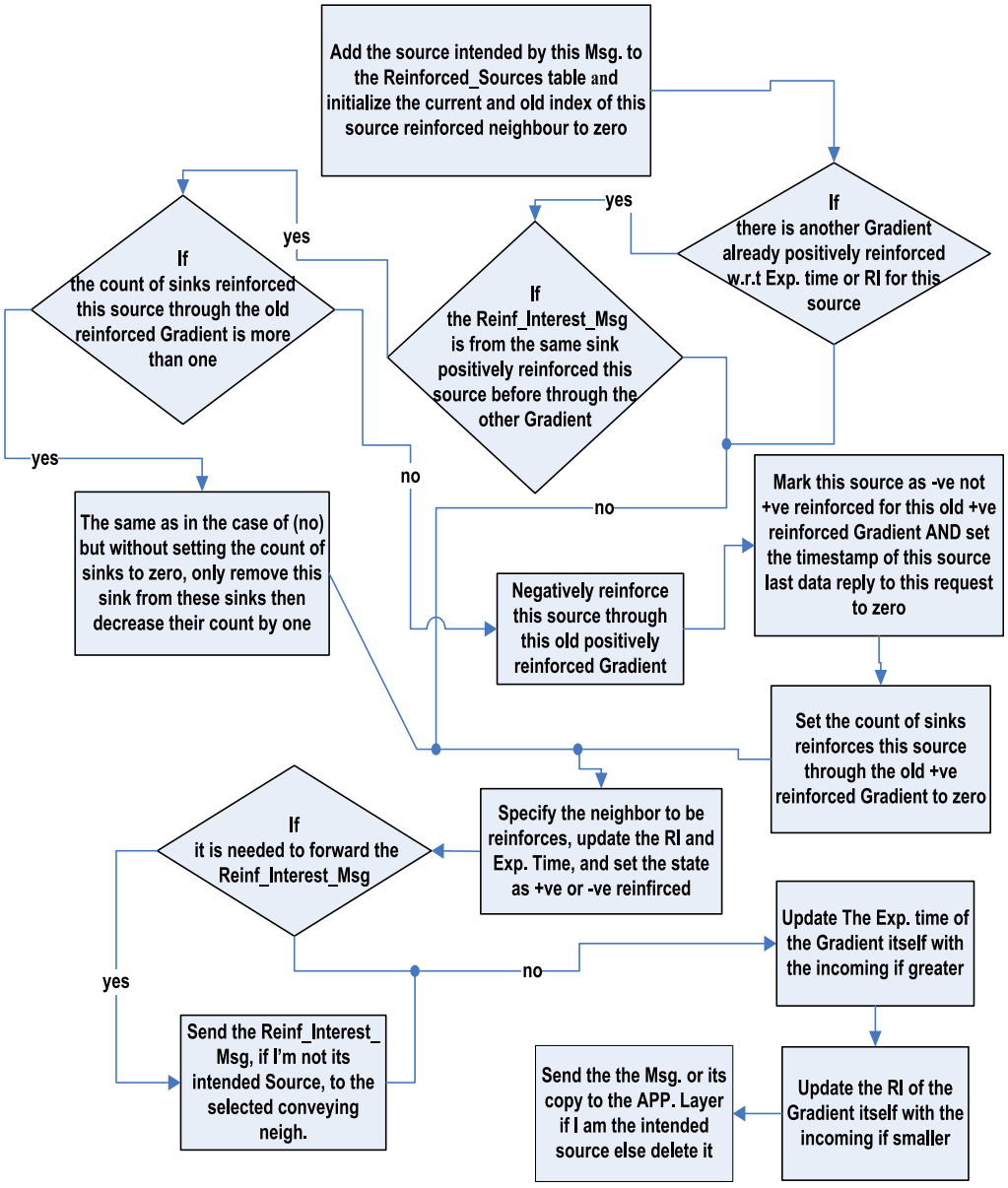

Figure 2 A part of Reinf_Interest_Msg processing by nodes in directed diffusion instantiation.

preferable to make this time period which used to test for sinistral sources adjustable to some specified different values according to an approximate prediction of the state of the network.

4. Loop-free local path repair implementation

With the previous implementation, all the possible probabilities to the sink for a path between it and a source equal only the count of its neighbors' neighbors that conveyed this source data. These paths' possibilities may be increased by making each node follows the sink behavior in sending the repair positive reinforcement message to the same old reinforced neighbor, but this causes the message to be sent to the sinistral node from its neighbor node because it does not know about its death, so the same old sinistral path will be established.

The previous problem will appear because of the absence of the knowledge of neighbor death from the node and the feeling of this node of losing a source data from a period compounds to the feeling of losing this source data of the other nodes which precede it in the old positively reinforced path from the precinct of the sink, in this case, each node as well as the sink remains to send the repair reinforcement to its dead neighbor unless it knows about its death, and it will not know about a neighbor death unless the neighbor informed it before its death. The local path repair for node death will help to solve this problem.

The local repair strategy proposed in this paper (Loop-free Local Path Repair (LLPR)) is implemented by using two different types of control messages, Death_Alarming and Choose_Another_Reinforced_Neighbor messages. The Death_Alarming contains a flag to indicate the type of the death whether it is a death because of energy exhaustion (Reason 0) which is

mentioned before or it is a death only with respect to some sources by exhausting all the neighbors that convey their data (Reason 1).

The node checks its energy level after receiving a Data_Msg, if it is below or equal to a threshold of energy suffices and slightly overflows its need for sending a Data_Msg and a Death_Alarming, it will begin firstly by broadcasting a Death_Alarming in its transmission range, if its energy level is above this threshold, it returns to check its energy level after each Data_Msg sent but against another energy threshold suffices and slightly overflows its need for sending two Data_Msgs and a Death_Alarming, then it follows the same just mentioned behavior.

Also after the reception of a Reinf_Interest_Msg, the node will not forward this message to another node if it found its energy level is below a threshold abundantly enough at least for sending the Reinf_Interest_Msg and receiving two Data_Msgs to allow it changes its state in the reinforced path for the intended source from New to Old, if its energy is below, it broadcasts a Death_Alarming.

The loop in the path causes the cut of the path at the end point of the loop, so for solving this problem, the node in a reinforced path for a certain source takes two states New or Old, in the beginning all the nodes are in the state New. Figure 3, describes New/Old state transition.

The remaining part of solving the problem of loops requires the Reinf_Interest_Msg to held two additional fields, one of them represents the count of hops from the sink of the Reinf_Interest_Msg initiator in the reinforced path for the intended source and the second represents the count of hops from the sink of the Reinf_Interest_Msg sender. The first field takes the value zero if the message is initiated by the sink for reinforcing the source or re- reinforcing it for path repair, and takes the value of the count of hops from the sink of any node if it initiates a path repair Reinf_Interest_Msg for this source. The flowcharts in Figures 4 and 5, represent the usage and processing of the Death_Alarming and the Choose_Another_Reinforced_Neighbor messages.

The Choose_Another_Reinforced_Neighbor is like in construction and dealing a Reason 1 Death_Alarming for the source intended by the Reinf_Interest_Msg, but it is sent only to the Reinf_Interest_Msg sender and it contains a new field to remind the sender about the count of hops from the sink of the message Initiator to use this count when resends the Reinf_Interest_Msg to another node.

5. Notes and critical assessment of LLPR

The local repair is used to solve the problems resulted from nodes' death because it is faster, gives better results, little in consuming energy, and little with respect to traffic congestion especially that the path was working what prevented its working is the death of one or more nodes. So there is no need to repair the path except in the place of node death. By that at the time of repair, the ability to determine whether the reading of this source is important for better determination of at least one target at the current time at the sink is lost, so a path repair can be made for a source its data is not important for the sink at the current time.

But the sink still needs this source data because although it is currently redundant, it is important for it at a later time if some sources sense the object dead and the reading of this source became needed for better target specification. At this time the data of this source is available at the sink, the sink has no need to repair the old cut path of this source and waits its data to come especially if no local repair was made to the redundant source, the sink will have no information about this source death or living, so it will remain to send the reinforcement to it until the test it performs after each negative data sending. Also if the source path is cut without repair, the source will remain to send data through the connected part of the cut path losing its energy and loses other nodes' energy without any use.

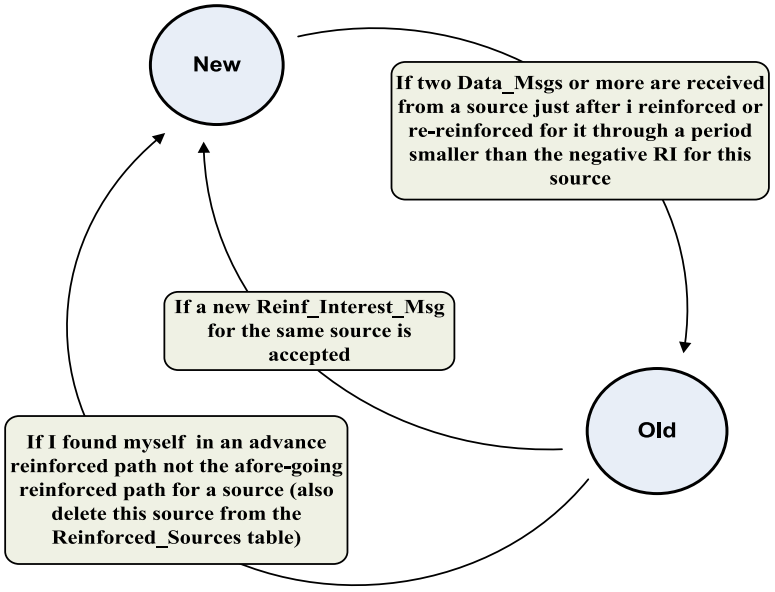

Figure 3 New/old state transition.

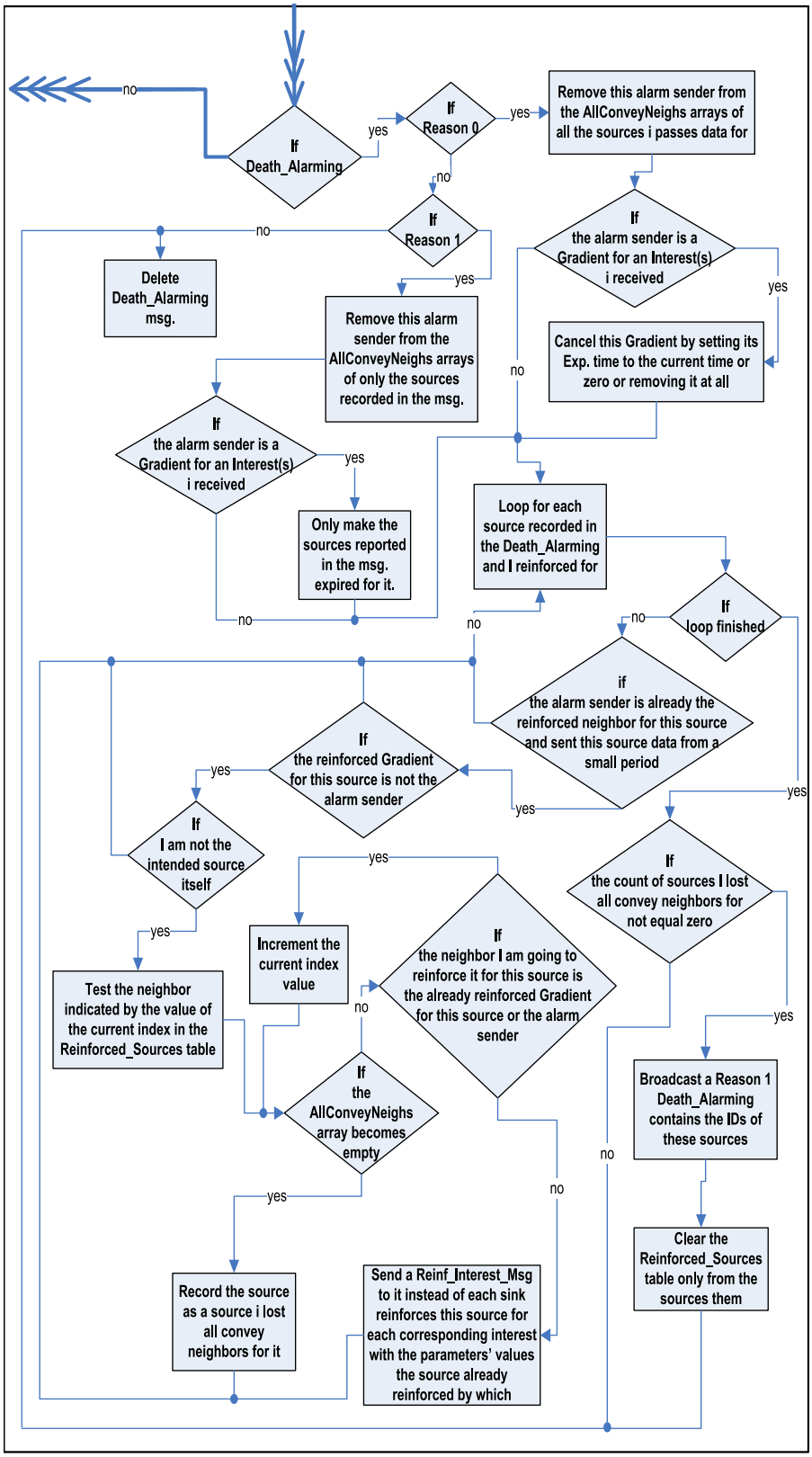

Figure 4 The processing of Death_Alarming message.

The interruption of a source data due to other factors such as node failure, node draft, and link cut due to bad weather conditions cannot be made by local repair because it costs the node high energy consumption and high storage capacity and increases its operation complexity. So, this task is still given to the sink with the same method it used except that if

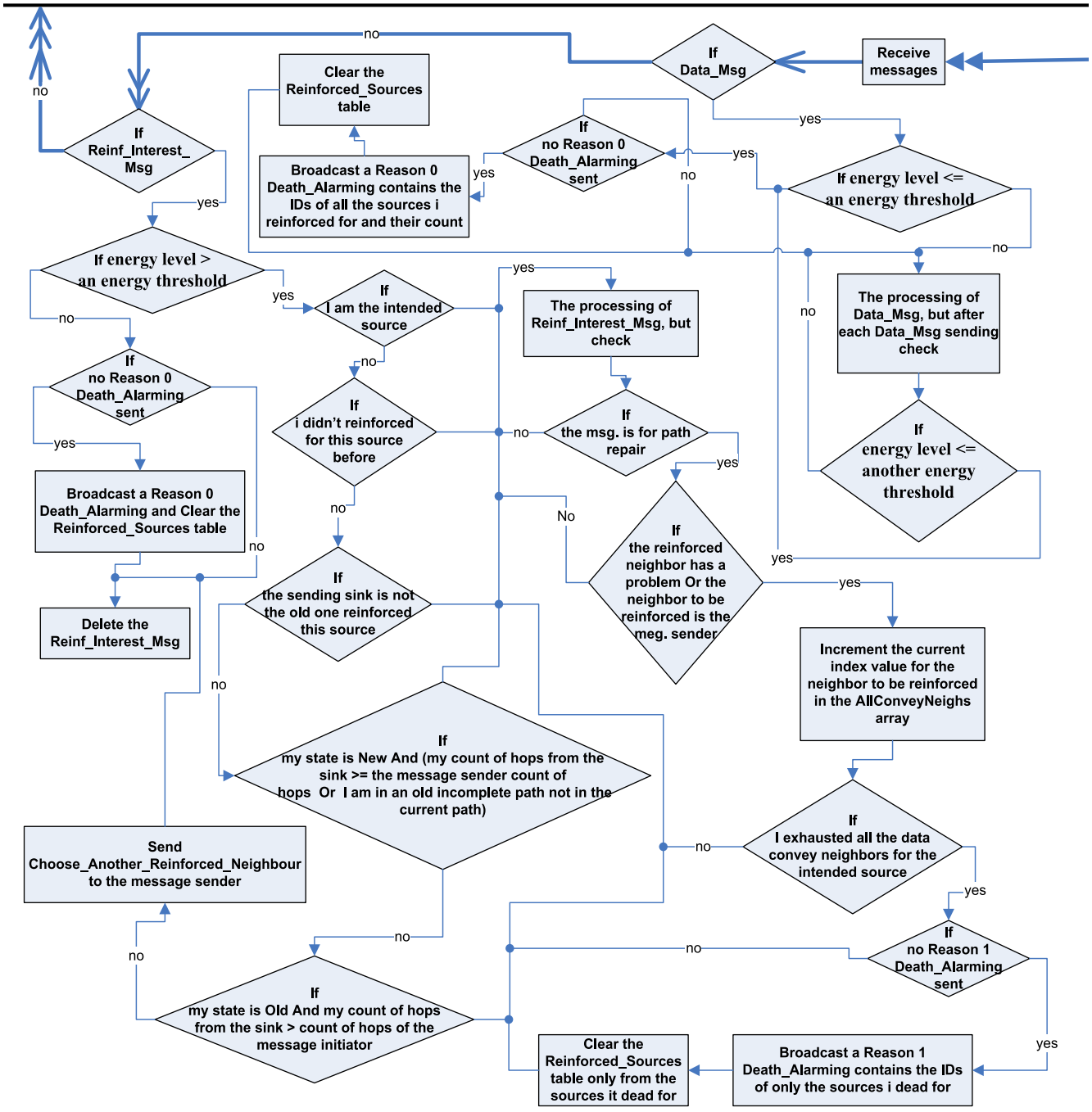

Figure 5 The method of death alerting.

the sink decided to not send reinforcement for a source due to the moving of the object outside its range, it will not send it a reinforcement in subsequent tests even if it assess that the object is now in the source range again because according to the test performed, the intersection of the two sensing ranges does not mean that the tested source sense the object because the object may be in the other side of the other source that actually senses it.

The local path repair implemented in this paper is a method for multiple sinks – multiple sources scenario and it:

- Specifies the point of path cut, this need the knowledge of death by the node caused the problem itself, this done by broadcasting a Death_Alarming message only when it is impending to die.
- Enhances the taken decision for repair, this is done by broadcasting the Death_Alarming, not sending it only to the node that will begin the repair process which is the Gradient of the node caused the problem. Upon the reception of the Death_Alarming, each node takes the appropriate decision based on the relation between it and the sender

node, so that, it will take the correct decision when performing a path repair later (for example, if the sender node is dead with respect to a specific source, it will not reinforce it for this source later).
– Gives the possibility to use in the new path not only the healthy part of the cut off path that precedes the cut point, but also, the other part of that path which follows the cut point or a portion of it.
– Needs no location information. Also, it does not need special control message sent for updating the relative locations in a path, the New and Old states and the count of hops make it is not necessary for the Death_Alarming to be propagated to nodes other than the nodes in the alarming node's range.
– Prevents loop formation to a point in the precedent or subsequent parts of the cut off path when repaired, where the loop in a path prevents data to reach the sink and/or aids in wasting possible paths from the loop end node. This is done only by using the two states New and Old, two additional fields in the Reinf_Interest_Msg, and another control message called Choose_Another_Reinforced_Neighbor which is not needed in a frequent manner.

Figure 6 shows the reinforced path of source 66, the ID of each node is written beside it. Before node 49 dies, it broadcasts a Death_Alarming message, so node 63 starts the path repair. Figure 7 shows the repair process, the count of hops from the sink of each node is put in a rectangle beside it. As shown in Figure 7, when node 30 is reinforced, in its first three trials to reinforce a neighbor, it selected nodes 58, 44, and 63 respectively which precede it in the path and they have different states, the LLPR prevents the formation of the loops which would formed due to this selection.

As this example shows, the repair is done locally and results in a complete path from sink to source, the repair decision is usually correct and energy-efficient, for example a New node will not reinforce node 49 if it is its neighbor because it knew about its death. Also, the repair results in a better usage of possible paths as the node will not change its reinforced node until it knows about its death. The loop is prevented up and down the repair initiator. The time from node 49 Death_Alarming broadcast to the arrival of the Reinf_Interest_Msg to the source 66 equals 0.043 s, which is surely will be much less than the used positive RI, this decrease the probability of wasting any important data packet. With respect to the delay, each

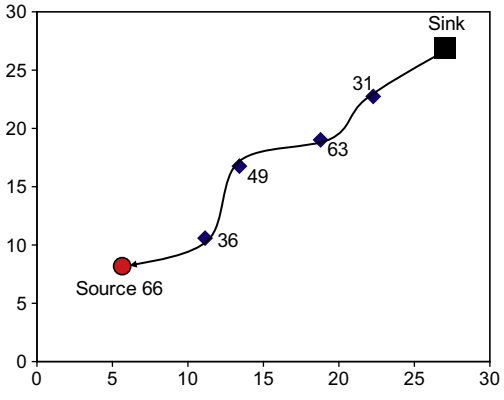

Figure 6 The reinforced path of source 66.

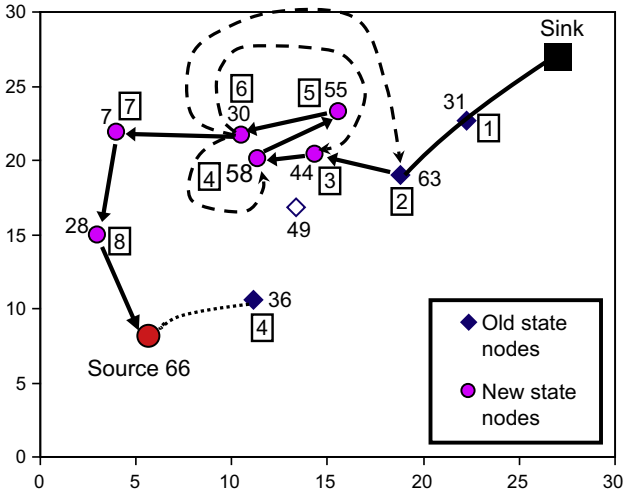

Figure 7 The path repair process of source 66.

node tries to do its best effort to decrease the length of the repaired path, but this is not given the priority.

To our knowledge, with the importance of addressing the looping paths problem, it is not studied sufficiently. Some techniques allow the data to travel in a loop, but prohibit infinite loops. Some others use the geographical location information to prevent loops formation. Other methods require a lot of special control messages and it may be propagated through the network. The authors of directed diffusion used the message cache to perform loop avoidance, but also they benefited from truncating the looping paths by negative reinforcement for resource savings. However, such loop removal is not always appropriate, specifically for some shared high-rate gradient maps with multiple sources and sinks.

6. Evaluation of directed diffusion instantiation

This section demonstrates the simulation setup including the employed general hypotheses, the used model for directed diffusion network evaluation, the performance metrics used in the evaluation, and the parameters' values.

6.1. The scenario assumptions

It is assumed that there is a number of sensor nodes are statically uniformly deployed in a two-dimensional square field, and the network is characterized by:

- There is one or more sinks are deployed inside the field.
- Sensor nodes are location-aware and non-rechargeable.
- The senor nodes can read and specify the value specific to each sensed object of a property of the objects, and all the sensors sense an object read the same value of its property, so that the objects of the same data type can be identified by their respective values of this property.
- The sensor nodes sense the object that enters their sensing ranges at the same time regardless of its distance from each node.
- For simplicity, it is assumed that the radio transmitter, radio receiver, and radio amplifier are the main energy consumers of a sensor node.

- The node can vary its transmission power depending on the distance to the receiver (as assumed in our previous work [12]).
- The Radio H.W. energy dissipation model [13] used is: to transmit a l-bit message over a distance d, the transmitter consumes

$$E_{Tx} = \begin{cases} l \times E_{elec} + l \times e_{fs} \times d^2, & d < d_o \\ l \times E_{elec} + l \times e_{amp} \times d^4, & d \geqslant d_o \end{cases} \quad (1)$$

And to receive that l-bit message, the receiver consumes

$$E_{Rx} = l \times E_{elec} \quad (2)$$

where e_{fs} and e_{amp} are the energies dissipated in the transmitter amplifier for either a free-space channel or a multi-path fading channel respectively. E_{elec} is the energy dissipated in transmitter and receiver electronics per bit.

- The signal propagation model used is the free space propagation model.
- All the sinks send the original Interest_Msgs with equal Initial RI but may be with different Exp. times.

6.2. The network model and parameters' values

The simulation runs were conducted using the discrete event simulator OMNeT++ [14] as the simulation platform to generate a network in $30 \times 30\ m^2$ area in which sensor nodes are distributed statically and uniformly. A simple tracking application is used to study the protocol, the simulation model is suitable for monitoring and tracking different number of objects from the same type or different types, these objects may be mobile or static, there may be more than one static sink in the monitored area and each sink can send a request for data different than the others.

For simplicity, in the simulation runs, it is assumed that there is only one sink node located at the point (27, 27) and it is assumed that it has infinite power and other resources. It sends an interest requests from the sensor nodes in the sub-monitored area [0, 10, 0, 10] the monitoring and tracking of four-legged-animals. At a certain point of time, an animal will enter this sub-monitored area followed by another animal, while the second animal remains in its place, the first animal moves with random steps in the sub-area until it exits it.

The simulation scenario can be described briefly in the following steps:

1. Each source node senses one or more objects and so each object is sensed by one or more sources. The sink receives the data from different sources and reinforces the minimum delay path to this source by sending a positive reinforcement interest message to the first neighbor node conveys this source data.
2. Accordingly this neighbor reinforces its neighbor node which first conveys to it the data from this source, and so on, until the positive reinforcement interest reaches the intended source (the processing used in simulation of the node's application layer for the received positive reinforcement interest is depicted in Figure 8).
3. The sink negatively reinforces its remaining neighbors which convey this source data later by sending to each of them a negative reinforcement interest message.
4. Each node receives this message resends it to all of its neighbors that conveyed this source data to it until the negative reinforcement interest reaches the intended source from its neighbors.
5. It is assumed that the sink uses the data come from the different sources (which are ID, x–y position, the data type of the sensed object, a specific property of this object identifies it from any other objects of the same data type, the distance of the object from the source, and the event timestamp) as soon as it reach in showing the place of the sensed objects in a screen; the more the sources sense the object, the more accurate the information about its place.
6. If only one source senses the object, the information that the sink can extract from this data when it receives it will be that, the object was existing at a certain time in an unspecific point on the periphery of a circle its center is this source and its radius is the distance reported by this source.
7. If two sources sense the object, the information at the sink will be more accurate, the sink will understand that the object at this time was at one of two locations represented by the two points of intersection of the two sensing circles corresponding to these two sources reported distances from this object.
8. If three sources sense the object, the sink can specify the location of the object at the reported time which is the point of intersection of the three sensing circles of the three sources, so the sink requires only the data of three sources at a certain time to specify accurately the location of the object.

The parameters' values used in simulation are stated on Table 1. The network is tested against different node densities to assess its behavior with respect to the performance metrics that stated in Section 6.3 when the initial deployed nodes' count is increased. The transmission and sensing ranges are chosen to be suitable to the sensed field size, such that the transmission range ensures connectivity between nodes with different node densities and in the same time be small enough to not increase transmission load on nodes. The sensing range is of course smaller than the transmission range and it is selected to achieve full area coverage with different node densities.

The sink is put inside the field to enable the nodes to communicate with it by the same transmission range by which they communicate with each other, but it is put in a point far from the sub-monitored area to evaluate the operation of the protocol as the messages propagate through a long distance through different count of hops.

The values of the interest messages' parameters are selected with the same ideas used in directed diffusion paper [1], the original interest RI equals 1 s and the value of positively reinforced RI is selected to be smaller. The negatively reinforced RI is set not to be small to not burden the network in sending a lot of redundant data and not to be long to fasten the repair decision correction as mentioned before. The reporting period is set suitable to the chosen initial energy value so that, it is enough to test the network behavior and also does not increase the simulation time.

6.3. Performance metrics

The directed diffusion implementation is evaluated with respect to the network lifetime, throughput, end-to-end delay, and loss percentage; these metrics are defined in simulation as follows:

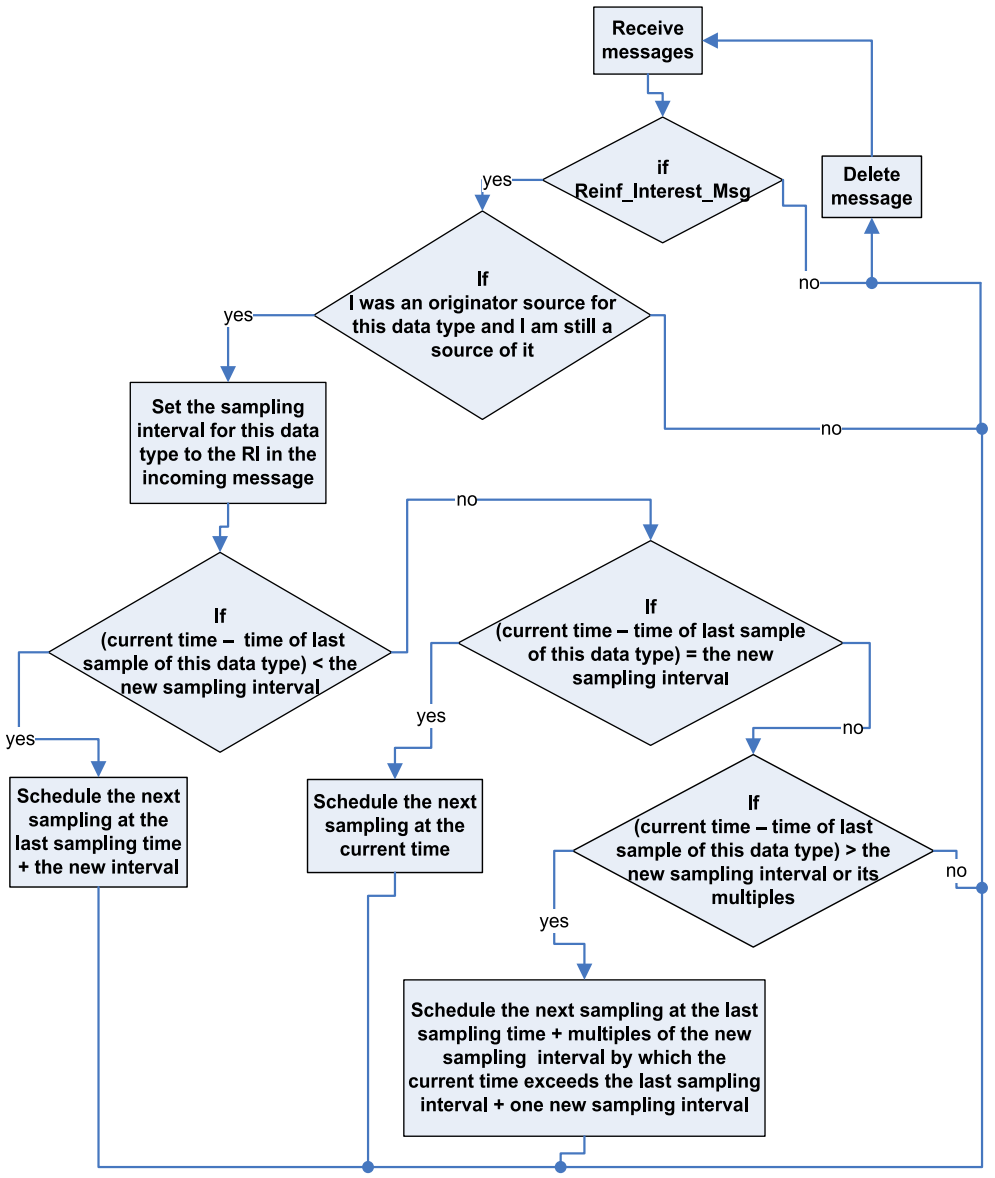

Figure 8 Node's application layer processing of the Reinf_Interest_Msg in directed diffusion instantiation.

(1) *Lifetime*: the time (measured in seconds) from the network deployment until the network partitioning, i.e., the time at which the sink no longer receives data due to cutting of all paths to the sub-monitored area, cutting of all paths to the sources in this sub-area, or the death of these sources themselves.

$$\text{Lifetime} = \text{network deployment time} - \text{network partitioning time} \tag{3}$$

(2) *Throughput*: the data received at the sink through the network lifetime divided by the lifetime (measured in (bits/s)).

$$\text{Throughput} = \frac{\text{number of bits received at the sink during the lifetime}}{\text{network lifetime in seconds}} \tag{4}$$

(3) *Packet loss percentage*: the percentage of the packets lost from all the packets sent by all sources during the network lifetime, taking into account that the same packet may be received at the sink through more than one path, to the count of packets sent by all sources during the lifetime.

$$\text{Packet loss percentage} = \frac{\text{all sources transmitted data packets count} - \text{count of packets received at the sink}}{\text{all sources transmitted data packets count}} \times 100\% \tag{5}$$

Table 1 Simulation parameters.

Parameters	Value
General parameters	
Network filed	(30, 30)
Nodes number	30–90
Broadcast range r	9 m
Sensing radius r_s	5 m
Sink position	(27, 27)
Initial energy	0.4 J
Data packet size	525 Bytes
Broadcast packet size	25 Bytes
E_{elec}	50 nJ/bit
e_{fs}	10 pJ/bit/m^2
e_{amp}	0.0013 pJ/bit/m^4
Threshold distance d_0	75 m
Original interest parameters	
Data type	Four-legged-animals
Area	[0, 10, 0, 10]
Reporting interval	1 s
Reporting period	1800 s
Positive reinforcement Interest parameters	
Data type	Four-legged-animals
Area	[0, 10, 0, 10]
Reporting interval	0.5 s
Reporting period	1800 s
Negative reinforcement interest parameters	
Data type	Four-legged-animals
Area	[0, 10, 0, 10]
Reporting interval	15 s
Reporting period	1800 s

(4) Average end-to-end delay the average of the ripening time of each information at the sink in each timestamp, i.e., the average time of the arrival of the third source reading for each data sample of an object averaged all over the sensed objects through network lifetime, and this time is computed by subtracting the sample timestamp from the arrival time of the last source reading maximized to three readings if the sink received three or more different readings about the object in this timestamp.

The time each packet containing information about a sensed object takes to travel from the source to the sink equals the count of the delays in the hops it crosses, and the delay of sending a packet from a node to its neighbor composed of the transmission delay, the propagation delay, and the queuing delay. The packet transmission delay equals the packet size divided on the data rate. The packet propagation delay equals the distance between the sender and the receiver divided on the speed of light, and the queuing delay of a packet equals the remaining time for the packet which is being transmitted at the entrance time of this packet in the transmission queue to finish transmission plus the count of the transmission delays of the packets waiting in the queue before this packet to be transmitted.

6.4. Results and analysis

The behavior of SPR can be analyzed in the following points:

- When the positively reinforced data of a source ceases reception at the sink, the sink sends another Reinf_Interest_Msg for the same old reinforced neighbor, which represents the start of the cut path, if the time of negative reinforced data reception of this source did not come yet or it came and it was found that this neighbor actually has healthy paths to the source and it delivers its data to the sink through them.
- The node that receives the repair reinforcement and finds that the remaining of the old reinforced path did not send data from a specified period, does not know if its neighbor is the dead node or another node in the remaining part of the path, so to compass this situation, it changes the whole remaining path by assuming that its neighbor is the dead node, and it departs from this sinistral path by changing it by changing the neighbor to reinforce with the hope that the new selected path warps around the problem infinitive.
- But it may be for all the nodes or it is usually for the sink neighbor nodes that the node's neighbor is not the dead node, so, it loses by choosing another neighbor not only one possible healthy path to the source through this neighbor, but more than one path through all the neighbor nodes of that neighbor, this causes the fast exhausting for the possible positive paths to the source available to each node, this effect manifests more when the nodal density is small.
- Also it is usually that this new selected neighbor selects the node that caused the problem in the old path or another dead node because it is its first neighbor that conveyed the source data to it, and it cannot determine if this neighbor is dead or alive. This richly done when the interval during which the sink sends the Reinf_Interest_Msg to a neighbor exhausted all of its possible positively reinforced paths increases, where the energy of some nodes during this interval is wasted due to wrongly sent Reinf_Interest_Msgs and lost Data_Msgs sent from the source.
- When a neighbor to the sink exhausts all of its possible positively reinforced paths, the sink continues to send the Reinf_Interest_Msg to it even after the test for the correctness of the last decision taken for path repair because it found that the source still sends negative data through this neighbor, and this case may continue for a long period until all the paths through this neighbor are actually cut or this neighbor dies unless the source itself dies, then the sink corrects the decision by sending the Reinf_Interest_Msg to another neighbor.

$$\text{End to end delay} = \frac{\sum_{\text{all sensed objects}} \left(\frac{\sum_{\text{all received data samples}} \text{arrival time of the data sample third source reading} - \text{data sample timestamp}}{\text{received data samples count}} \right)}{\text{count of sensed objects}} \quad (6)$$

- So from the previous talking, we can deduce and say that:
  - The smaller the nodal density, the smaller the number of nodes' neighbors, the smaller the count of a node's possible positively reinforced paths, the more the losses of data packets due to the fast exhausting of these paths, this causes the early surceasing of the source positive data although the source may be still alive.
  - The larger the nodal density, the larger the traffic load, the larger the probability of early and larger nodes' death, the more the losses of data packets, especially that the number of sources in this case is larger and the death of a node may affect more than one source, i.e., loses more than one source data, so the loss percentage increases especially that the sources themselves die earlier, and although the count of a node's possible positively reinforced paths increases, but also the probability of a node to select a dead node in each new path increases.
  - The larger the nodal density, the smaller the period in which the sink sends wrongly the Reinf_Interest_Msg for path repair to a neighbor exhausted all of its possible positively reinforced paths, the smaller the losses if the next neighbors the sink then reinforced them are alive, still have possible positively reinforced paths, and the remaining paths they constitute has no dead nodes.

A mixture from these rational logic deductions makes the shape of the loss percentage curve to be as shown in Figure 9, the curve with square markers corresponding to the legend "SPR", it approximately does not have a specific stable behavior, but the prevailing epithet of the curve is the increase with node density increase and also the difference between the curve points not considered to be large, so we can say the meant effect of the increasing in the count of possible positively reinforced paths for each node and the probable decreasing of the period in which the sink sends message to wrong neighbor when the nodal density increases is mitigated by a repugnant effect due to earlier nodes' death.

Loop(s) which is mentioned before may be formed in the reinforced path which aids in increasing loss percentage; when the node received the Reinf_Interest_Msg finds that the neighbor it will go to send the Reinf_Interest_Msg to it is itself the node sent it this Reinf_Interest_Msg, it does not select it and selects the next neighbor, but it cannot detect if the selected neighbor is not the immediate sending node but it is an earliest sending node for this message in the anterior part of the path, so it sends to it again the message it sent causing the loop, the earliest sending node reacts to this by selecting another neighbor than it selected before which also greatly aids in the fast exhausting of the possible paths to the source available to each node. This loop may be repeated in one path to the same node or to others and this is more apparent in big node densities.

It is obvious from Figure 9, the curve with triangle markers corresponding to the legend "LLPR", that the usage of LLPR strategy reduces the loss percentage by a big value about 92.69%, as it specifies the point of the problem, makes the path repair local, aids in better, faster, and accurate repair decision, and it makes the best use of all the healthy possible positive paths.

The lifetime in case of SPR as shown in Figure 10, decreases with node density increase because of the increase in the traffic load accordingly earlier sources' death and the earlier death of the critical nodes around the sink and around the sources. But it should be noted that, if the partitioning of the network in larger node densities is faster than in smaller densities in a working network, in smaller densities the network may be deployed partitioned, some nodes are separated from the others which may isolate the sink from targets' sources.

As shown in Figure 10, the lifetime in case of LLPR is decreased by about 29.79%. It is not a disadvantage, because the smaller lifetime in case of LLPR is better exploited in sending data than in case of SPR and this is a reason of lifetime reduction where the nodes are burdened with sending a large amount of data messages lost by SPR. Also LLPR lifetime reduction is rebounded to the burden in sending and receiving additional control packets.

Due to the decreasing behavior of lifetime and the unstable behavior of loss percentage, the throughput also does not take a stable behavior but it can be said that it is approximately constant or tilts to declining as shown in Figure 11, but it should be noted that, because the count of sources increases with node density increase the sink benefits from the smaller lifetime in the case of larger node density more than the larger lifetime in case of smaller node density; the average count of the precious specification of targets' position (three sources readings for the target distance in the same timestamp needed

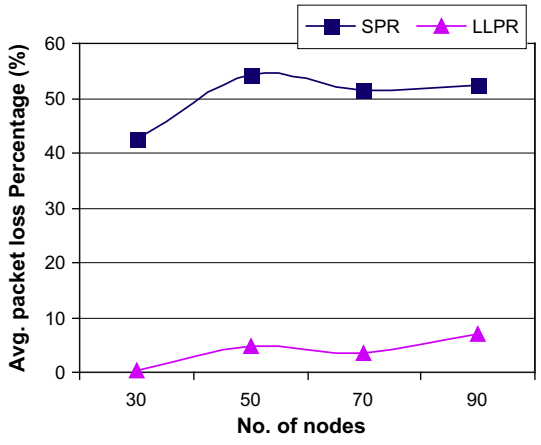

Figure 9 Average packet loss percentage vs. number of nodes.

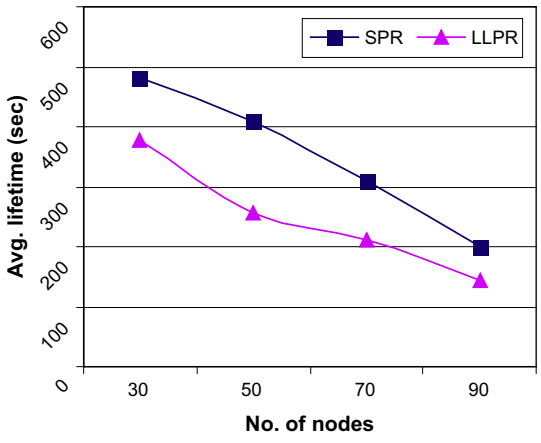

Figure 10 Network lifetime vs. number of nodes.

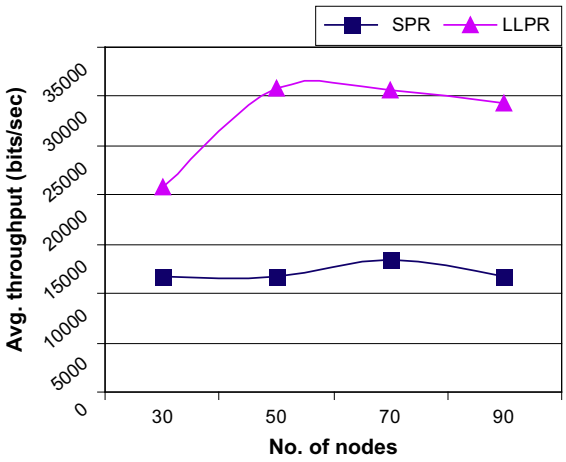

Figure 11 Average throughput vs. number of nodes.

to accurately specify a target position) increases in larger node density.

The LLPR throughput is increased as shown in Figure 11 by about 55.31% because as mentioned before the smaller lifetime is exploited in sending data, not in sending control packets for path repair with incorrect decision up to a healthy complete path is established.

As shown in Figure 12, the average end-to-end delay increases with the node density increase. The sink positively reinforces a source by sending the Reinf_Interest_Msg to the first neighbor conveys its data and so does every node receives the Reinf_Interest_Msg, so, a low delay path has a small count of hops is firstly constituted between the sink and the source may approximately have the same count of hops with node density increase.

While in Reinf_Interest_Msg sending for path repair, the keenness of nodes for the new path only to move around the point(s) of problem in the old cut path makes them do not give a big concern to the length of the new path (the count of hops constitute the path), it is correct that the node selects the immediate next neighbor to the old one delivered to it the source data, but this results in longer path than the old one and the number of hops in the repair path increases with the

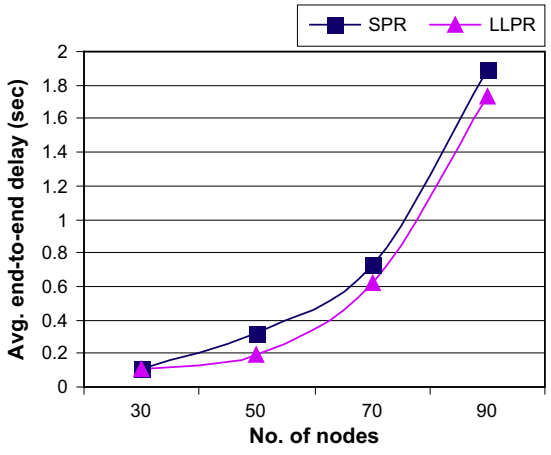

Figure 12 Average end to end delay vs. number of nodes.

increase in node density, especially that, not all the available neighbors to a node lead to healthy paths, so the healthy path finally it constituted may be very long.

The average end-to-end delay in case of LLPR is decreased as shown in Figure 12 by about 14.06%; this is predictable and resides in that the new path is not searched from the beginning and the repair is made only at the problem point. Also the larger available possibilities of positive paths prevent from the obliging of small count of possible paths may be long and prevent losing shorter paths.

7. Conclusions and future work

This paper studies by simulation directed diffusion as a data gathering and dissemination paradigm for wireless sensor networks. Under some conditions and assumptions and based on the basics of directed diffusion explained in its paper, an instance from this paradigm is proposed to simulate the directed diffusion wireless sensor network for tracking applications for the purpose of understanding its operations, evaluating its performance, and auditing on some of the probable design mistakes which cause performance degradation or undesirable results.

This paper also compares the directed diffusion performance in two cases, the first is the repair of the cut paths by the sink (SPR) and the second is the local repair of cut paths due to nodes' death (LLPR), in this case the nodes are given the responsibility of detecting, reporting, and dealing with the nodes' death, so that the point of cutting is specified accurately then the path repair is performed at this point. Also LLPR works on preventing path loops formation. LLPR improves the Directed Diffusion behavior with respect to loss percentage, throughput, and end-to-end delay by about 92.69%, 55.31%, and 14.06% respectively, while the lifetime is decreased by about 29.79%.

The design space of directed diffusion is very wide; there may be a large number of its instantiations for each different application type by varying the methods used in positive and negative reinforcements, data aggregation, path repair, and interest propagation. So, its design can bear a lot of modifications and techniques to improve the performance and exploit its tradeoffs, so it could be said that the directed diffusion performance can be improved not only to prevent or reduce the data messages loss during network lifetime and increase throughput for example, but to achieve design aims, it is aspired to reach them, and may be in the same protocol instance such as long increasing or constant lifetime and smaller decreasing or constant end-to-end delay with increasing nodal density to improve accuracy.

References

[1] Intanagonwiwat C, Govindan R, Estrin D. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In: Proc 6th MobiCom; 2000. p. 56–67.

[2] Kannammal KE, Purusothaman T. New interest propagation mechanism in directed diffusion protocol for mobile sensor networks. Eur J Sci Res 2012;68(1):36–42.

[3] Hady AA, Abd El-kader SM, Eissa HS, Salem A, Fahmy HMA. Internet and distributed computing advancements: theoretical frameworks and practical applications. USA: IGI Global; 2012. p. 212–46 [chapter 9].

[4] Dargahi F, Rahmani AM, Samadabadi R. Novel algorithms and techniques in telecommunications and networking. Netherlands: Springer; 2010, p. 477–481 [chapter 82].
[5] Kannammal KE, Purusothaman T. Evaluation of directed diffusion protocol for mobile sensor networks. Int J Eng Sci Technol 2010;2(6):2272–7.
[6] Perwaiz N, Javed MY. A study on distributed diffusion and its variants. In Proc 12th international conference on computers and information technology (ICCIT '09); 2009. p. 44–9.
[7] Akyildiz IF. Wireless sensor networks, series in communications and networking. New Jersey: John Wiley & Sons Ltd.; 2010.
[8] Misra S, Woungang I, Misra SC. Guide to wireless sensor networks. London: Springer-Verlag; 2009.
[9] Zheng J, Jamalipour A. Wireless sensor networks: a networking perspective. New Jersey: John Wiley & Sons, Inc.; 2009.
[10] Boukerche A. Algorithms and protocols for wireless sensor networks. New Jersey: John Wiley and Sons; 2009.
[11] Sohraby K, Minoli D, Znati T. Wireless sensor networks: technology, protocols, and applications. New Jersey: John Wiley and Sons; 2007.
[12] Mohammad El-Basioni BM, Abd El-kader SM, Eissa HS, Zahra MM. An optimized energy-aware routing protocol for wireless sensor network. Egypt Inform J 2011;12(2):61–72.
[13] Heinzelman WB, Chandrakasan AP, Balakrishnan H. An application specific protocol architecture for wireless microsensor networks. IEEE Trans Wirel Commun 2002;1(4):660–70.
[14] Varga A. Omnet++ discrete event simulation system. The Technical University of Budapest, Department of Telecommunications (BME-HIT); 2005. <http://www.omnetpp.org/omnetpp/doc_details/2105-omnet-32-win32-binary-exe>.