A Self-organizing Concurrent Global Information Dissemination Scheme for Sensor Networks

Ehssan Sakhaee, Naoki Wakamiya, Masayuki Murata
Graduate School of Information Science and Technology
Osaka University
Osaka, Japan
{sakhaee, wakamiya, murata}@ist.osaka-u.ac.jp

Abstract—**In this paper we propose a distributed and self-organizing scheme for establishing efficient energy-saving concurrent data dissemination for wireless sensor networks. The proposed protocol is aimed at applications where all sensor nodes need to provide their current data concurrently and periodically back to the sink, whist having an option to provide instantaneous individual node data upon sensing. Hence, the scheme provides a hybrid time-driven and an event-driven mechanism. The scheme initially establishes the hopcount of sensor nodes from the sink and identifies the highest hopcount nodes (HHN) of the sensor network. An application scenario is the periodical recovery of the global extremity value back to the sink by only using the HHNs for initiating the broadcast, which inherently traverses all sensor nodes due to the nature and position of such nodes. This approach can hence effectively retrieve sensor data from all nodes concurrently in its target application. Furthermore, the scheme provides a platform for other data-centric protocols to efficiently and concurrently disseminate data from all nodes in the network.**

Keywords—self-organization; energy-efficient data dissemination; sensor network routing.

I. INTRODUCTION

Information dissemination and reporting is an integral function of sensor networks. Due to the primary limitation of a sensor network, namely its lack of energy, it is important to design protocols that can effectively report data back to the sink without overtly exhausting the energy of the nodes. In this paper we propose a routing protocol for sensor networks which, upon establishing the current hopcount of all nodes from the sink, identifies the *highest hopcount nodes* (HHNs) of the network in a self-organizing manner. The HHNs are special (source) nodes from which their broadcast is able to efficiently reach all nodes within the network. Once the HHNs identify themselves, they will periodically broadcast an information packet which is updated and forwarded by intermediate nodes back to the sink. This efficient mechanism can save a considerable amount of energy by eliminating the need for individual node broadcasts and flooding of the network, whilst presenting concurrent sensor data or some data which relies on all sensor data (such as max/min/average) back to the sink. This is especially important for nodes closer to the sink as in traditional sensor network routing protocols, such nodes tend to lose the most amount of energy, due to their extensive forwarding (relaying) of messages from other nodes in the network (positioned further away from the sink) to the sink [11]. The primary application of this scheme is for sensor networks where simultaneous or concurrent data from all sensor nodes are required to determine a value, or parameter. A direct example of this is a sensor network, where the sensors report the current extreme value (e.g. highest temperature/pressure) of a field periodically back to the sink. The proposed routing mechanism which is initiated by the HHNs is dubbed as the *highest hopcount node-initiated* (HHNI) routing. Additionally an event-driven mechanism is integrated into the distributed and self-organizing HHNI algorithm, dubbed Sensor Node Initiated (SNI) routing, which allows individual sensors to be triggered when a threshold is reached, and report their current values to the sink prior to the periodic reporting of HHNI algorithm.

Why is the approach self-organizing? In self-organizing systems such as those in biological and emergent systems, all individual organisms follow identical set of rules, however behave and react differently depending on *feedback* from their immediate environment (local information) [3]. However, an emergent property or favorable global property emerges from such simple local interactions. Similarly, in the proposed protocol, each node follows the same set of rules, and establishes its status (e.g. HHN/normal node, and hopcount to sink) based on the feedback from its own neighbors (locally) without any global knowledge. The emergent properly of the resulting network is efficient concurrent dissemination of field data back to the sink. We believe that self-organization is important in sensor networks in order to provide 1) cost-effectiveness and quick setup of an efficient network without predefining node states, 2) reduction or elimination of centralized control, and 3) freedom from tinkering individual node behavior pre/post-deployment.

The proposed scheme may be used in conjunction with other routing protocols to more efficiently report any form of data back to the sink. Hence it is believed that the scheme can also be adapted to and or adopt other data aggregation techniques used in some data-centric protocols. However in this paper the application scenario is based on extreme value (maxima/minima) reporting of data to demonstrate the basic mechanism of the protocol and its effectiveness in its target application.

II. BACKGROUND

In [1] a scalable and energy-efficient routing scheme is proposed for large-scale wireless sensor networks. This protocol, however requires the knowledge of location of nodes throughout the network. Other such well-known protocols include the Greedy Perimeter Stateless Routing (GPSR) [2] and its extensions [4] used for sensor networks, which also requires the knowledge of position of nodes for making routing decisions. However, in many cases, knowledge of sensor locations is costly and often infeasible for wireless sensor networks. Previous energy-efficient routing schemes do not tackle the fundamental issue of redundant broadcasting. Probabilistic forwarding is proposed [5] [6] which reduce the energy consumption of nodes by eliminating flooding. However, the scheme does not immune the network against redundant (initial) broadcasts by individual nodes, which wastes precious node energies, often, unnecessarily. Some of the protocols also require flooding of the network to obtain costs of paths, which is another energy-depleting mechanism used in traditional networks. Geographic and Energy Aware Routing (GEAR) [7] also tries to minimize energy usage by using geographic location of nodes. However, this also requires the position of nodes.

With regards to data-centric protocols, Sensor Protocol for Information via Negotiation (SPIN) [8] was one of the first attempts to reduce redundant and duplicate data from circulating. However the advertisement in SPIN will again use more energy than is required for the target application of obtaining extreme value of a field, although SPIN may be more suitable for more sophisticated data aggregation applications. Another issue with SPIN is that intermediate nodes which are *not interested* in the data may prevent the data to be delivered to a sink which is far away from source [9]. Directed Diffusion [10] is yet another data-centric protocol which aims at *diffusing* interest data through attribute-value pairs for the required data. The protocol requires querying nodes for interest data, which may incur additional and unnecessary energy consumption when applied to the application target of extreme-value discovery of a network, or where the data to be obtained requires or is a function of *all* sensor nodes within the network. Furthermore, hierarchical data aggregation techniques have been proposed such as LEECH [12] and HEED [13], which aggregate node data at specific nodes called clusterheads. However, the cluster-based approaches are again not suitable for the application of extreme-value discovery of a network. Other generic data aggregation techniques are outlined in [14].

III. THE PROPOSED SCHEME

A. Highest Hopcount Node-Initiated (HHNI) Routing

The main aim of the proposed protocol is to minimize redundant broadcasts by broadcast initiation of certain nodes that allow subsequent traversal of all nodes in the network in an efficient way, instead of individual nodes broadcasting their values. Such nodes are termed the *highest hopcount nodes* HHNs which initiate broadcast and other intermediate nodes simply update the message and forward it on towards the sink. This idea is illustrated in Fig. 1. Additionally, redundant rebroadcasts are minimized by *selective-forwarding*, which involves performing simple comparisons of consecutive messages at each node, via locally saved information and only forwarding “selected” messages. Moreover, a *random waiting* mechanism is integrated at each node for avoiding identical message broadcasts/rebroadcasts which further minimizes redundant rebroadcasts. In the initial model, it is assumed all nodes are equally spaced from each other. We also assume the sink functions separately to the rest of the sensor nodes in the network; however it uses the same transmission range as the other nodes within the network. In Fig. 1, the transmission range of nodes is set to one-hop. The black node (0.0) represents the sink, the grey nodes represent sensor nodes where nodes with hopcounts less than 9 represent the intermediate (ordinary sensing/forwarding) nodes and the node with hopcount equal to 9 representing the HHN node.

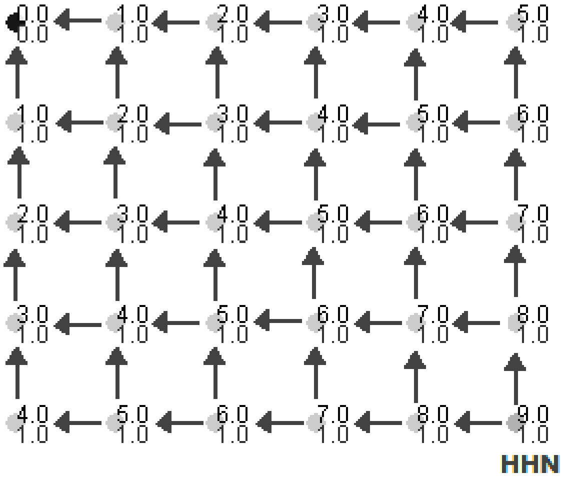
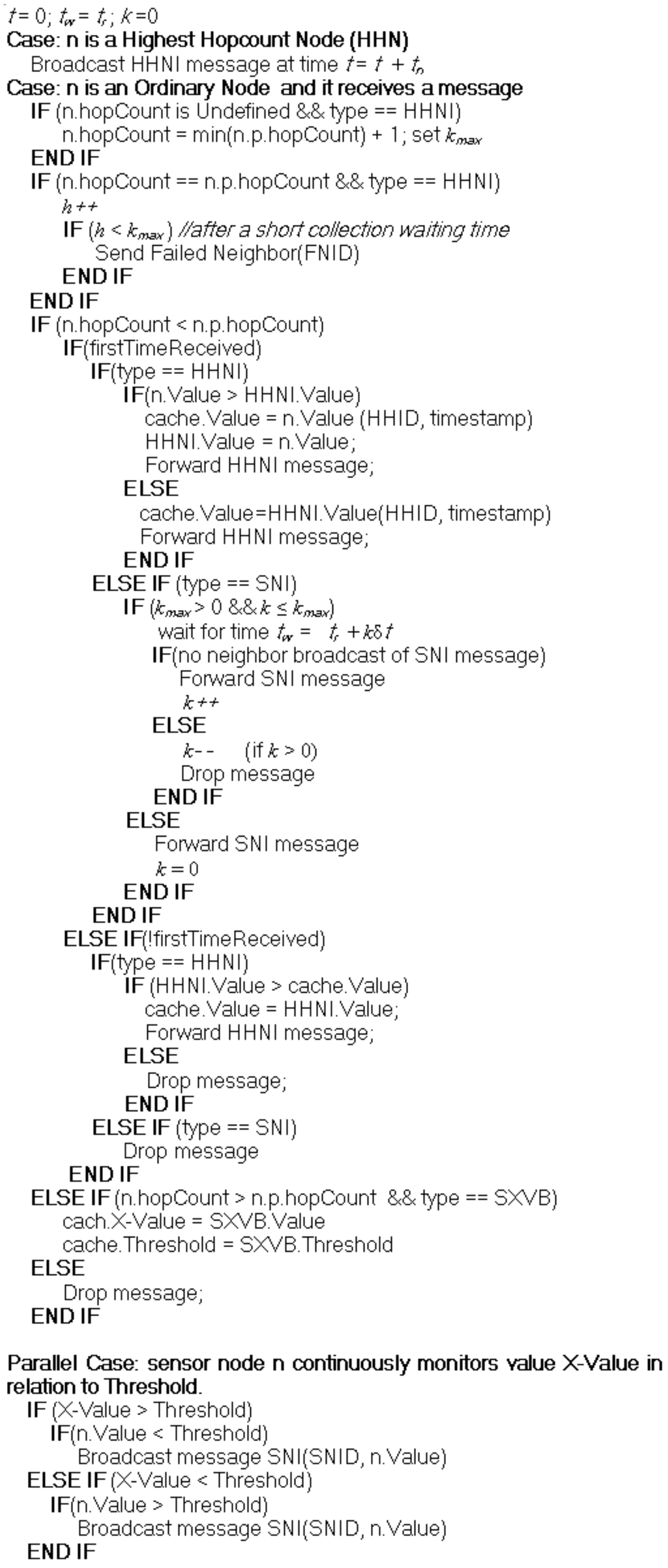


Fig 1. Network topology.

In order for nodes to discover their relative hopcount to the sink and for HHNs to establish their HHN status dynamically, we use the following *initialization* algorithm.

1. Sink broadcasts an “S” message to neighbors.
2. Neighbors of sink update their *hopcount to sink* to 1 and broadcast a “1” hopcount message to their own neighbors.
3. An intermediate node whose hopcount is yet *undefined*, sets its hopcount value as the minimum among all received hopcount messages plus 1, and broadcasts this hopcount to its neighbors.
4. Nodes set their k_{max} variable, which is the number of neighbor nodes with the same hopcount. This variable is set after overhearing neighbors broadcasting their hopcount messages.
5. When a node **does not** receive a hopcount message with a *larger* hopcount than its own, it assumes it is the *HHN* of the field.



```
$t = 0$; $t_w = t_r$; $k = 0$
Case: n is a Highest Hopcount Node (HHN)
  Broadcast HHNI message at time $t = t + t_n$
Case: n is an Ordinary Node  and it receives a message
  IF (n.hopCount is Undefined && type == HHNI)
    n.hopCount = min(n.p.hopCount) + 1; set $k_{max}$
  END IF
  IF (n.hopCount == n.p.hopCount && type == HHNI)
    $h$++
    IF ($h < k_{max}$) //after a short collection waiting time
      Send Failed Neighbor(FNID)
    END IF
  END IF
  IF (n.hopCount < n.p.hopCount)
    IF(firstTimeReceived)
      IF(type == HHNI)
        IF(n.Value > HHNI.Value)
          cache.Value = n.Value (HHID, timestamp)
          HHNI.Value = n.Value;
          Forward HHNI message;
        ELSE
          cache.Value=HHNI.Value(HHID, timestamp)
          Forward HHNI message;
        END IF
      ELSE IF (type == SNI)
        IF ($k_{max} > 0$ && $k \le k_{max}$)
          wait for time $t_w = t_r + k\delta t$
          IF(no neighbor broadcast of SNI message)
            Forward SNI message
            $k$++
          ELSE
            $k$--     (if $k > 0$)
            Drop message
          END IF
        ELSE
          Forward SNI message
          $k = 0$
        END IF
      END IF
    ELSE IF(!firstTimeReceived)
      IF(type == HHNI)
        IF (HHNI.Value > cache.Value)
          cache.Value = HHNI.Value;
          Forward HHNI message;
        ELSE
          Drop message;
        END IF
      ELSE IF (type == SNI)
        Drop message
    END IF
  ELSE IF (n.hopCount > n.p.hopCount  && type == SXVB)
    cach.X-Value = SXVB.Value
    cache.Threshold = SXVB.Threshold
  ELSE
    Drop message;
  END IF

Parallel Case: sensor node n continuously monitors value X-Value in
relation to Threshold.
  IF (X-Value > Threshold)
    IF(n.Value < Threshold)
      Broadcast message SNI(SNID, n.Value)
  ELSE IF (X-Value < Threshold)
    IF(n.Value > Threshold)
      Broadcast message SNI(SNID, n.Value)
  END IF
```

Fig. 2. Unified HHNI/SNI algorithm.

The hopcount initialization is propagated from the sink to the edge of the network and this leads to the minimum initialization overhead by making each node broadcasts a hopcount message only once. A node, which receives a hopcount message with a larger hopcount than itself, does not update its hopcount, because it sets its hopcount value as the minimum plus 1. Hence it also does not rebroadcast a hopcount message upon receiving one with a larger hopcount than itself.

After the initialization is complete, all nodes will know their hopcount to sink and HHNs have identified themselves. The hopcount message also contains the periodic time interval for which broadcast should be initiated. Once the HHN nodes are identified, they would proceed with *periodic* broadcasts of information regarding their environment (e.g. temperature, pressure etc). Upon the initiation of the broadcast of the HHN, the nodes which have a lower hopcount than their parent node (the node that the broadcast is heard from) will continue forwarding the broadcasted messages until it reaches the sink. This is the basic principle of HHNI routing. In addition to this mechanism, we can further reduce redundant rebroadcasts by aggregating data or depending on the application storing the extreme (e.g. maximum/minimum) known and discovered values at each forwarding node, and having such a node check consecutive messages (identical timestamps) and only forward the ones that have higher extreme values than those stored at the node. We classify this as *selective-forwarding* of HHNI or HHNI-SF. At this point other data aggregation and comparison techniques may also be incorporated.

Reporting in the proposed scheme can occur via two means: HHNI broadcasting, and Sensing Node Initiated (SNI) reporting (also an inherent broadcast). The SNI uses its own message type which bypasses the comparison algorithm of the HHNI scheme. The SNI simply needs to report the specific value of a sensing node to the sink, and is not interested in the values of other nodes in the field. The unified algorithm is shown in Fig. 2. In the figure, n is the current node, p is the immediate parent of the current node (from which the message was received), t is the current time (initialized to zero), t_n is the interval between each periodic broadcast, t_w being the waiting time before broadcast forwarding (priority waiting), t_r a random time between zero and t_{max}, where t_{max} is the maximum random waiting time allowed, and δt an incremental period. k is an integer variable which is incremented each time a node broadcasts, and decremented each time a node overhears a broadcast. h is the number of overheard (HHNI) broadcasts, and is incremented by one each time an identical timestamp HHNI broadcast of the same hopcount as n is heard. If h does not reach k_{max}, it indicates that one or more neighbor nodes have not responded, implying a failed neighbor node, and so a Failed Neighbor message is sent to the sink indicating the ID of the failed node (FNID) which did not respond. The value of k_{max} is determined for each node during the initialization phase described earlier. This is directly proportional to the number of neighbors present which have the same hopcount as n. Value corresponds to the data value, which is assumed a single value in this case. The cache is the local memory of the node where values and corresponding identifiers and timestamps of received messages are stored.

We note that when an intermediate node receives a SNI message type, it will wait for a random time before broadcasting to see if its neighbors broadcast the same message. Each time a message is broadcasted by a node, the random waiting time is incremented by δt and decremented by this value when a node overhears a broadcast by its neighbor. This is to first give priority to nodes which have not previously broadcasted, and then conceive priority once neighbors have broadcasted. This prevents nodes of the same hopcount that hear the message from broadcasting the same message avoiding redundant broadcasting whilst providing a more distributed broadcasting mechanism to reduce individual node energy exhaustion. This random time waiting mechanism is only used for SNI reporting, as in HHNI, information from all nodes is required. In the target application network, the number of sources is equivalent to the number of nodes in the network (excluding the sink).

The sink is also able to set the current network's extreme value (X-Value) and threshold of all nodes, which is used by the node to monitor its current value for SNI reporting. If a node receives a sink extreme value broadcast (SXVB) message, it will set the X-Value and Threshold. Once these two values are defined at a node, the node compares its sensed data (value) against these values. This usually happens upon changes in the sensed data's value. The X-Value simply determines whether the SNI reporting occurs when the node's currently sensed data falls above or below the threshold. For instance, if we wish a node to report its value when its current value falls below the threshold, we set the threshold below the X-Value, otherwise we set the threshold above the X-Value if we wish to have the node report its value when its current sensed value rises above the threshold. Finally, the sink collects and evaluates the messages, obtaining/extracting the desired value or extremity and broadcasting this to all sensor nodes (for HHNI) or simply using the value obtained without further input into the system (SNI).

Pure HHNI reporting simply forwards the message on without comparing values. However, this acts as a clear platform for other potential data-centric, aggregation, and negotiation protocols which are in existence [9] [14] to be able to effectively and efficiently retrieve values from all nodes concurrently. Pure HHNI is used when the values of all nodes are to be collected and studies, e.g. for scenarios where the average of all sensor nodes' values need to be obtained, or where statistical evaluation of sensor values needs to be studied. However, it should be noted that in the target application in this paper, we are not considering data negotiation in our protocol as used in some of the previous data-centric protocols such as [8] [10] as we assume there is no overlap [8] in the sensing regions. Hence each node uniquely obtains the value of its own unique region. Despite this assumption, the proposed protocol may be adaptable to data negotiation schemes for the application of disseminating all sensor node data concurrently and efficiently.

B. Message Types

The message types are shown below. The Type field specifies whether the message is of type HHNI or SNI, since nodes will react differently depending on the type of message. The HHN ID, (HHNID) is the unique ID of the HHN initiating the broadcast. This ID can be set statistically (predefined), or can be generated dynamically, however it must be unique. For instance the ID can be a function of the hopcount to sink and the node's residual energy at the time of instantiation. For the HHNI broadcast, the Value field corresponds to the maximum value so far, whereas in the SNI message, the value corresponds to the value at the specific node with ID of SNID (hence this field is not updated) as the message is propagated. In the SXVB message, the X-Value is the extreme value of the network at the instance of data collection at the sink, extracted from the HHNI or SNI broadcast/reporting. The Threshold field is used to monitor the value of the current node. If the threshold value falls below or above this value, then a SNI report is triggered to immediately report the current node value to the sink. The X-Value is used at each node to determine whether triggering occurs when the value of node falls below or above the threshold.

HHNI broadcast

< Type: HHNI><HHNID><Value><Hopcount><Timestamp>

SNI reporting

< Type: SNI><SNID><Value><Hopcount><Timestamp>

SXVB broadcast

<Type: SXVB><X-Value><Threshold>

Note that the Hopcount field is updated with the current node's hopcount at each node before the message is forwarded on (provided it is forwarded). A node can distinguish between a HHNI, SNI, and SXVB broadcast by simply checking the Type field of the message.

C. In case of Node Failure

Failure of HHNs: When a node which was previously a *neighbor* of a HHN does not hear a HHN broadcast for some period t_m where t_m is given by $t_m = mt_n$ and where m is the number of missed broadcast intervals and t_n is the interval between broadcasts, it will claim the status of a HHN and begin its function as a HHN (i.e. performing periodic broadcasts). This follows the self-organizing rule.

Failure of non-HHN Nodes: Failed nodes can be physically replaced by new nodes if necessary, in which case the new nodes can update their hopcount-to-sink as they overhear or forward the messages initiated by the HHN. This is also effective when relative node positions of normal nodes change, in addition to the new deployment of new nodes in the network.

Moreover, in addition to the self-organizing rules mentioned, a "backup" *centralized* mechanism may be used, where when a periodic broadcast is not heard at the sink (failure of HHNs), a new H/HHI procedure can take place. However this is not necessary and acts as a backup to the self-organizing procedure. Hence the network can function purely without any feedback from the sink, provided the H/HHI occurs just once in order to initialize the network.

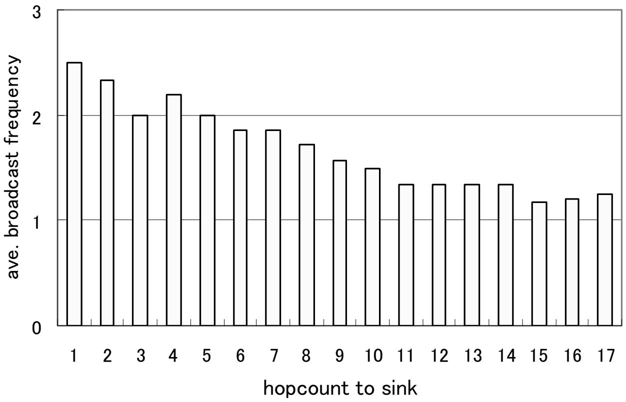


Fig. 3. Broadcast frequency of individual nodes for 100 nodes.

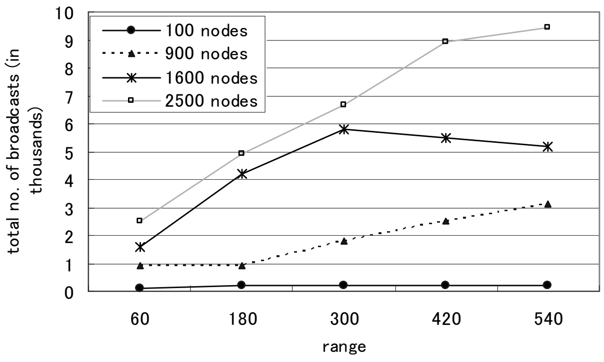


Fig. 5. total number of broadcasts with varying transmission range.

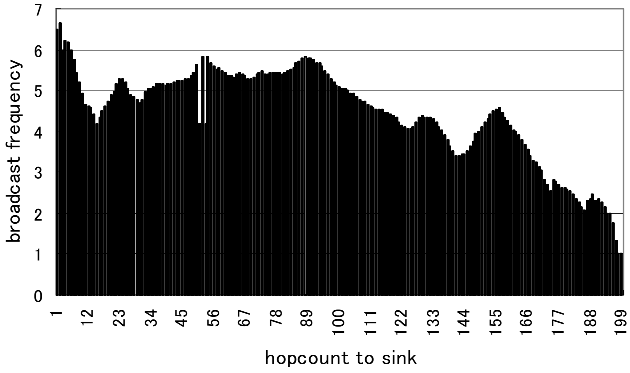


Fig. 4. Average broadcast frequency of individual nodes for 10,000 nodes.

0.0	1.0	1.0	2.0	2.0	3.0	3.0	4.0	4.0	5.0
1.0	1.0	1.0	2.0	2.0	3.0	3.0	4.0	4.0	5.0
1.0	1.0	2.0	2.0	2.0	3.0	3.0	4.0	4.0	5.0
2.0	2.0	2.0	2.0	3.0	3.0	3.0	4.0	4.0	5.0
2.0	2.0	2.0	3.0	3.0	3.0	4.0	4.0	4.0	5.0
3.0	3.0	3.0	3.0	3.0	4.0	4.0	4.0	5.0	5.0
3.0	3.0	3.0	3.0	4.0	4.0	4.0	5.0	5.0	5.0
4.0	4.0	4.0	4.0	4.0	4.0	5.0	5.0	5.0	6.0
4.0	4.0	4.0	4.0	4.0	5.0	5.0	5.0	6.0	6.0
5.0	5.0	5.0	5.0	5.0	5.0	5.0	6.0	6.0	6.0

Fig. 6. Formation of HHNs with increasing transmission range.

IV. SIMULATIONS

Simulations were performed to evaluate the effectiveness of the proposed scheme. The application platform is for reporting the highest value sensor node back to the sink. Node values are randomly chosen, between zero and one. Each node contains a random value. The distance between nodes is set to a fixed 50 units. A rectangular topology is used similar to the one shown in Fig.1. In all simulations the HHNs are the initiator for broadcasting. Furthermore the highest value node is reported back to the sink (simulating HHNI-SF algorithm). Fig. 3 and Fig. 4 show the frequency of broadcast (number of times a node performs a broadcast, including initial broadcast, and forwarding) by individual nodes for a field of 100 nodes and 10,000 nodes respectively, for the first round of reporting initiated by the HHN (node with hopcount of 199).

In Fig. 3, the transmission range is set to 60 units (equivalent to one-hop), so that nodes only broadcast to their one-hop neighbors. Fig. 5 shows the total number of broadcasts when varying the transmission range for different network sizes (characterized by the number of nodes). The effect of increasing the transmission range here has the same effect as increasing the node density, e.g. by reducing the distance between sensor nodes, so that the sensor node can reach more than one-hop neighbors. From Fig. 3, it can be seen that the maximum average broadcast frequency of 2.5 belongs to nodes one-hop away from the sink. As node hopcount from sink increases, the number of broadcasts tends to decrease. Similarly in Fig. 4, the same trend is observed, although the average broadcast frequency is higher for the network with 10,000 nodes. Although the general trend is a decrease in broadcast frequency with increasing hopcount, the actual frequency also depends on the relative location of extreme value nodes, which causes the downstream nodes to perform additional broadcasts. In Fig. 5, as we increase the transmission range of sensor nodes, the total number of broadcasts increase. The best performance is achieved when the transmission range is equivalent or close to a one-hop transmission. This is the case regardless of the network size. It is interesting to note that as the transmission range increases, so do the number of HHNs. This is shown in Fig. 6 for the case where the transmission range is increased to two-hops. This is due to the nature of the algorithm, which impels nodes that do not overhear a broadcast greater than their own hopcount, to elect themselves as HHNs. The result of this is the increase in the number of broadcasts as visible in Fig. 5. In the figure, there are a total of 16 HHNs. We note that although the highest hopcount in the field is six hops, some nodes which have a hopcount of 5 have elected themselves as HHN. Although these nodes do not possess the highest hopcount *globally*, they are *locally* the highest hopcount nodes as the actual HHNs (hopcount of 6) are not within their range. We note that the local HHNs are necessary for traversing *all* nodes within the network.

V. CONCLUSION AND FUTURE WORK

In this paper we propose a scheme for efficiently reporting extreme values of a field-based sensor network, by first identifying the highest hopcount nodes of the network, which their broadcasts are capable of penetrating all nodes within the network efficiently. Consequently this approach is able to report extreme values of the network to the sink as efficiently as possible. The principle idea may also be integrated into future protocols depending on the application. For example, cost-based sensor network routing protocols may effectively use the highest-hopcount-node-initiated (HHNI) broadcasting scheme presented in this paper to more efficiently perform routing in sensor networks based on various metrics such as energy, delay, and hopcount.

In future work, the effect of mobility of nodes should be considered to support more dynamic and mobile networks. Furthermore, since in this paper it is assumed that the links are bidirectional, the effects of unidirectional links can also be considered in future research. Although the simulated application scenario involves a uniformly distributed topology, random layout networks may also take advantage of the proposed scheme. However the limitations (if any) of the proposed scheme on randomly distributed sensor networks should be investigated in further detail.

ACKNOWLEDGEMENTS

This research was supported in part by the “Global COE (Centers of Excellence) Program” of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

REFERENCES

[1] G. Lukachan, M. Labrador and W. Moreno, “Scalable and energy-efficient routing for large-scale wireless sensor networks,” in Proc. ICCDCS’06, Mexico, Apr. 2006.

[2] B. Karp, and H. Kung, “Greedy perimeter stateless routing,” in Proc. 6th ACM MobiCom’00, Boston, MA, Aug. 2000.

[3] S. Kauffman, “At home in the universe: the search for the laws of self-organization and complexity,” AIP Press, 1995.

[4] J., Chen, Y., Guan, U. Pooch, "Customizing a geographical routing protocol for wireless sensor networks," in Proc. ITCC’05, vol.2, no., pp. 586-591 Vol. 2, 4-6 Apr. 2005.

[5] K. Chen, Y. Qin, F. Jiang, Z. Tang, “A probabilistic energy-efficient routing (PEER) scheme for ad-hoc sensor networks,” in Proc. SECON’06, vol. 3, pp. 964-970, Sept. 2006.

[6] R. C. Shah, and J. Rabaey, “Energy aware routing for low energy ad hoc sensor networks,” in Proc. IEEE WCNC’02, Orlando, FL, Mar. 2002.

[7] Y. Yu, D. strin, and R. Govindan, “Georgraphical and energy-aware routing: a recursive data dissemination protocol for wireless sensor networks,” UCLA Com. Sci. Dept. tech. rep., UCLA-CSD TR-010023, May 2001.

[8] W. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” in Proc. ACM MobiCom’99, Seattle, WA, Aug. 1999.

[9] J. N. Al-Karaki, A. E. Kamal, “Routing techniques in wireless sensor networks: A survery,” Wireless Communications Magazine, Dec. 2004.

[10] C. Intanagonwiwat, R. Govindan and D. Estrin, “Directed diffusion: A scalable and robust communication paradigm for sensor network,” in Proc. 6th ACM MobiCom’00, Boston, MA, Aug. 2000.

[11] P. Cheng, C. N. Chuah, X. Liu, “Energy-aware node placement in wireless sensor networks,” in Proc. IEEE GLOBECOM’04, vol. 5, pp. 3210-3214, Dec. 2004.

[12] W. R. Heinzelman, “Application-specific protocol architectures for wireless networks,” Ph.D. thesis, Massachusetts Institute of Technology, Jun. 2000.

[13] O. Younis and S. Fahmy, “HEED: a hybrid, energy-efficient distributed clustering approach for ad hoc sensor networks,” IEEE Trans. Mobile Computing, vol. 3, no. 4, pp. 366-379, Dec. 2004.

[14] R. Rajagopalan, P. K. Varshney, “Data-aggregation techniques in sensor networks: a survey,” IEEE Communication Surveys and Tutorials, vol. 8, pp. 48-63. 4th Quarter, 2006.