WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. (2008)

An energy-efficient self-organizing global extremity reporting scheme for sensor networks

Ehssan Sakhaee*,†, Naoki Wakamiya and Masayuki Murata
Graduate School of Information Science and Technology, Osaka University, Osaka, Japan

Summary

In this paper we propose an energy-efficient self-organizing global extremity reporting scheme for wireless sensor networks. The proposed scheme assists applications of periodic reporting of extreme values (such as maximum or minimum temperature/pressure) across a wireless sensor field, back to the sink. Furthermore, an event-driven counterpart is supplied for individual sensor nodes to supply their instantaneous sensed values back to the sink, once queried. The targeted sensors initially establish their relative distances to the sink in regards to number of hops, whilst the highest hopcount nodes (HHNs) from the sink identify themselves. The broadcast initiation of the HHNs have the ability to penetrate all nodes within the network towards the sink, and hence obtain the extreme value of the entire network in an efficient manner. This is due to the relative position of these special nodes within the network. Furthermore, the scheme does not require nodes to possess location information of themselves or other nodes, avoiding the need for the global positioning system (GPS) or other location-aware methods. Simulation results show the effectiveness of the proposed protocol in its target application. In particular, the advantage of HHN-initiated broadcasting can be seen in both uniformly and randomly distributed topology networks. Copyright © 2008 John Wiley & Sons, Ltd.

KEY WORDS: self-organization; energy-efficient data dissemination; sensor network routing

1. Introduction

The general goal of a sensor network is efficiently and reliably retrieving sensed information in the network by using the least amount of network resources as possible. 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 is able to retrieve the extreme values of a network in an efficient manner. The protocol initially establishes the current hopcount of all nodes from the sink in a distributed and self-organized manner, and finally identifies the *highest hopcount nodes* (HHNs) of the network. The HHNs are special nodes at the highest parts of the network, and furthest away from the sink, and from which their broadcast to their one-hop neighbors is able to efficiently reach the entire nodes in the network which are located downstream to the HHNs and in the direction of the sink. Once the HHNs identify themselves, they

---

*Correspondence to: Ehssan Sakhaee, Graduate School of Information Science and Technology, Osaka University, Osaka, Japan.
†E-mail: sakhaee@ist.osaka-u.ac.jp

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 data which relies on all sensor data (such as max/min/average) back to the sink. This mechanism also minimizes excessive energy loss of 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 [1]. 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) [2]. 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 property 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.

2. Background

In Reference [3] 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) [4] and its extensions [5] used for sensor networks, which also requires the knowledge of the 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 [6,7] 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) [8] also tries to minimize energy usage by using geographic location of nodes. However, this also requires the position information of nodes.

With regards to data-centric protocols, Sensor Protocol for Information via Negotiation (SPIN) [9] 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 [10]. Directed Diffusion [11] is yet another data-centric protocol which aims at *diffusing*

DOI: 10.1002/wcm

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 Reference [14]. An energy-efficient routing protocol for sensor networks was also proposed in [15]. Furthermore, a self-organizing and self-managing mechanism for sensor networks with satellites was proposed in [16].

3. The Proposed Scheme

3.1. 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 HHNs which initiate broadcast and other intermediate nodes simply update the message and forward it on towards the sink. This idea is illustrated in Figure 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 Figure 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 represents the HHN.

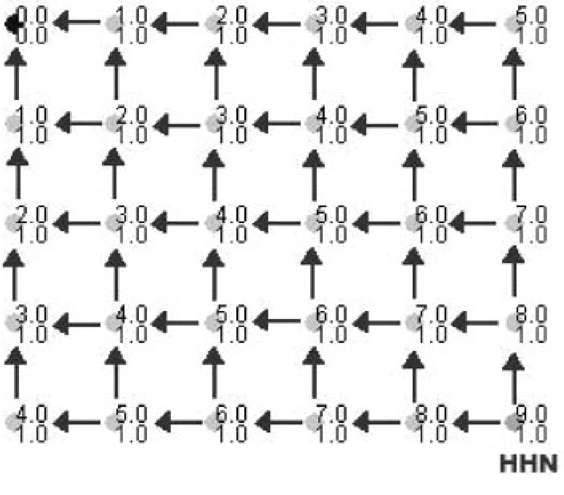


Fig. 1. Network topology.

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 *hopcount 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 itself to be the *HHN* of the field.

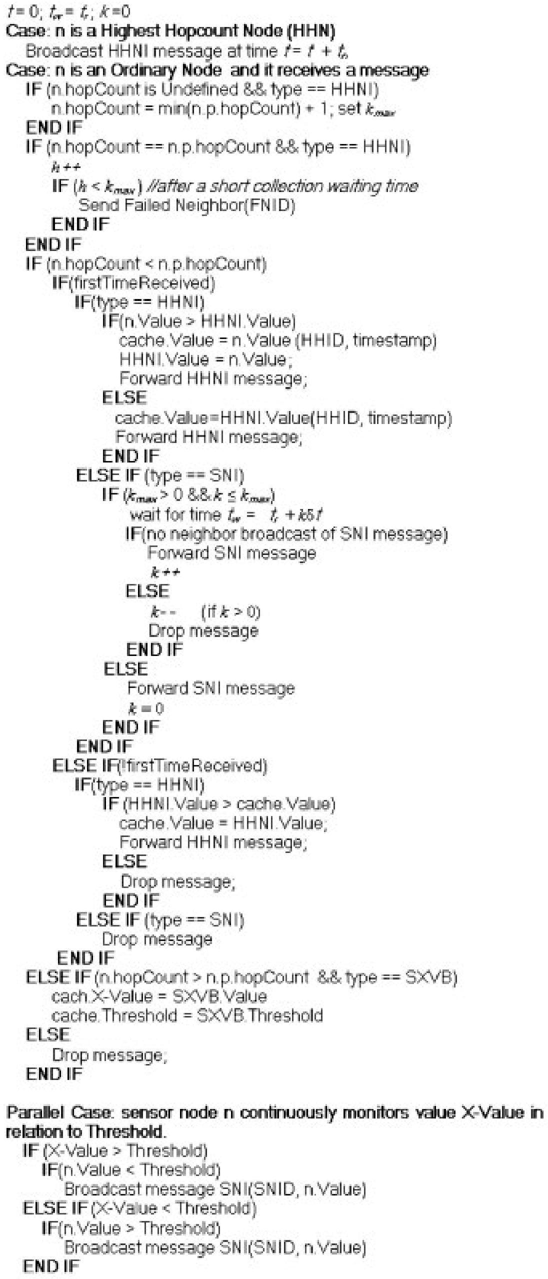
The hopcount initialization is propagated from the sink to the edge of the network and this leads to the minimum initialization overhead by making each node broadcast a hopcount message only once. A node, which receives a hopcount message with a larger hopcount than itself, does not update its hopcount, 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 hopcount 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 HHNs are identified, they would proceed with *periodic* broadcasts of information regarding their environment (e.g., temperature, pressure, etc). Upon the initiation of the

DOI: 10.1002/wcm

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 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 Figure 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 hopcount 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.



```
$t = 0$; $t_w = t_r$; $k = 0$
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.

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

DOI: 10.1002/wcm

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).

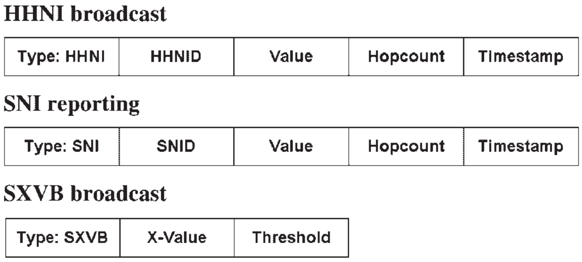
Pure HHNI reporting simply forwards the message on without comparing values. However, this acts as a clear platform for other potential data-centric, aggregation, and negotiation protocols which are in existence [10,14] to be able to effectively and efficiently retrieve values from all nodes concurrently. Pure HHNI is used when the values of all nodes are to be collected and studied, 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 References [9,11] as we assume there is no overlap [9] 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.

3.2. Message Types

The message types are shown in Figure 3. 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. 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.

3.3. 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.



HHNI broadcast

Type: HHNI	HHNID	Value	Hopcount	Timestamp

SNI reporting

Type: SNI	SNID	Value	Hopcount	Timestamp

SXVB broadcast

Type: SXVB	X-Value	Threshold

Fig. 3. Message types and their fields.

DOI: 10.1002/wcm

Failure of non-HHNs: 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 hopcount initialization 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 hopcount initialization occurs just once in order to initialize the network.

4. Simulations

Simulations were performed using the Java programming language 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 square topology is used similar to the one shown in Figure 1. In all the simulations, the highest value node is reported back to the sink (simulating HHNI-SF algorithm). Figures 4 and 5 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. In both figures, the transmission range is set to 60 units, so that nodes only broadcast to their closest one-hop neighbors only. We note that 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 Figure 4, it can be seen that the maximum average broadcast frequency of 2.5 belongs to nodes one-hop away from the sink.

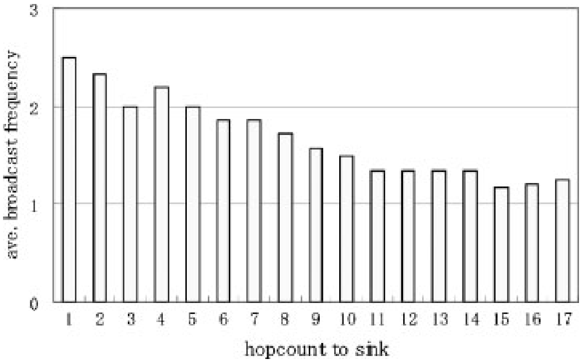


Fig. 4. Broadcast frequency of individual nodes for 100 nodes.

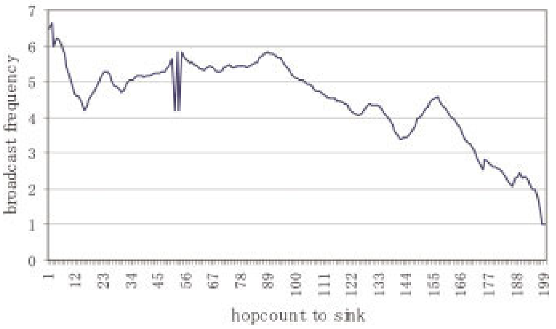


Fig. 5. Average broadcast frequency of individual nodes for 10 000 nodes.

As node hopcount from sink increases, the number of broadcasts tends to decrease. Similarly in Figure 5, 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. It is interesting to note that as the transmission range increases, the number of HHNs vary. An example of this is shown in Figure 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. 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 HHNs 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. Figure 7 shows the effect of increasing range on the number of HHNs in the network. In the figure, the number of

DOI: 10.1002/wcm

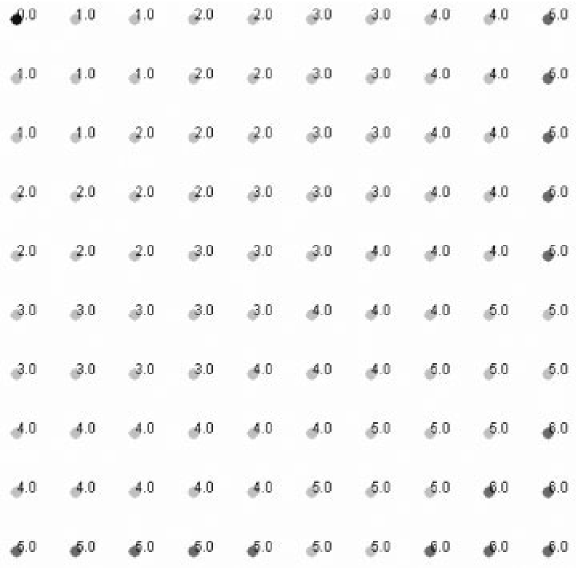


Fig. 6. Formation of HHNs with increasing transmission range.

HHNs generally increases with increasing transmission range, however there are visible peaks and troughs in the number of HHNs. Particularly we note the sudden increase in the number of HHNs at the range of 1740–3435 m. To illustrate this phenomenon, we must first analyze and understand how the range is related to the number of HHNs. According to the target topology we can create a simplified model which describes this relationship. Figure 8 demonstrates the effect of increasing transmission range on the number of HHNs in the network. In Figure 8(a), the transmission range is such that three HHNs are produced. As the transmission range is increased as of Figure 8(b), the number of HHNs increases to 10. A further increase in the transmission range results in only one HHN being produced as shown in Figure 8(c), a sudden drop of the number of HHNs. A slight increase however results in a sharp increase in the number of HHNs as shown in Figure 8(d). Similary, in Figure 7, before the critical ranges of 1740–3435 m, there are much fewer HHNs (resembling the case of Figure 8(c). As the transmission range is just slightly increased, the situation mirrors that of Figure 8(d) and the number of HHNs suddenly increases from 511 to 1515. The range of 1740 m is where the transmission range of the sink (and all nodes) bypasses half of the network diameter, and all nodes following the second hopcount message do not hear hopcount messages greater than themselves, as the second hopcount message broadcast has reached all the rest of the nodes within the network (region 2 in Figure 8(d)). In Figure 7, at the point of 3425 m, the transmission range bypasses the network diameter, and hence all the 2500 nodes become HHNs, as they do not hear hopcount messages greater than themselves. In this case the sink is one hop away from all within the network. Although the number of HHNs does not seem to have an effect on the total number of broadcasts, its knowledge may be used for other purposes, such as synchronization of broadcast (it is easier to

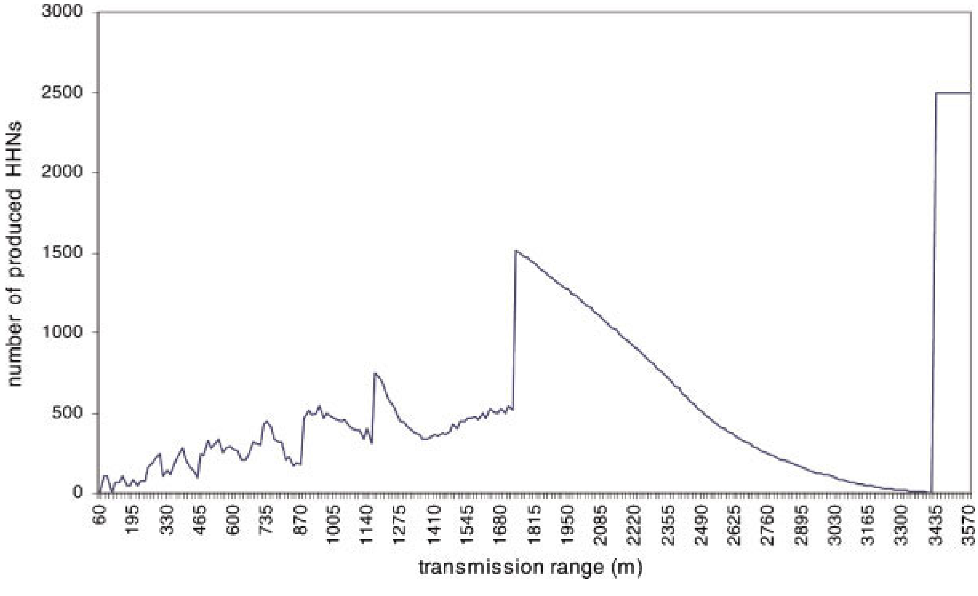
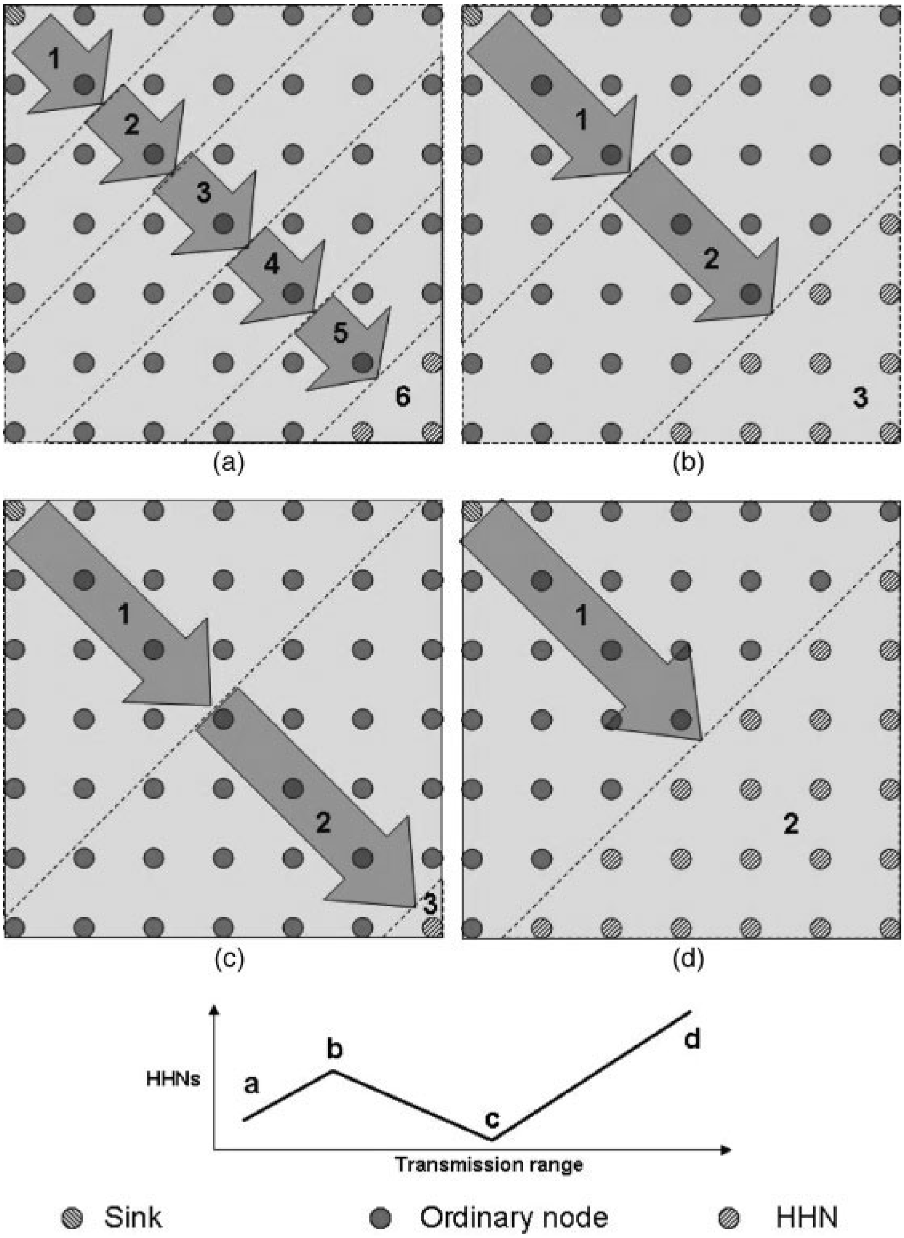


Fig. 7. Number of HHNs produced by increasing the transmission range.

DOI: 10.1002/wcm



(a) (b)

(c) (d)

Fig. 8. The effect of increasing range on the number of HHNs.

synchronize fewer HHNs together than a large number of them).

Figures 9 and 10, demonstrate the benefit of using HHNs for broadcast initiation. For this purpose, we use the sensor node energy consumption model in Reference [17] that uses 100 pJ/bit/m^2 for the transmitter amplifier.

The figures show the total energy consumption of the network in broadcasting sensed data, for 2500 and 10 000 nodes, respectively, for both HHNI and non-HHNI broadcast. For the non-HHNI broadcast nodes randomly send their data towards the sink, by also following all the rules as HHNI algorithm except the initiation of broadcast by the HHNs within the network (pure traditional flooding/gossiping schemes are not considered as they scale poorly in relation to both HHNI and non-HHNI methods). Figure 9 shows the total network energy consumed in broadcasting after a single report from 2500 nodes to the sink. The figure shows that as the transmission range is increased the

DOI: 10.1002/wcm

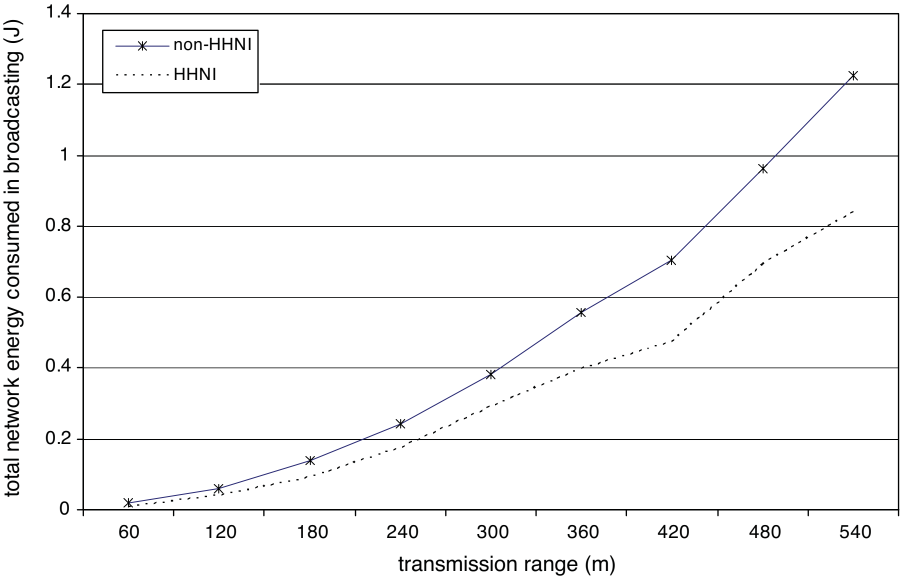


Fig. 9. Energy consumption versus transmission range for HHNI and non-HHNI routing for 2500 nodes after a single report.

total network energy consumption increases for both HHNI and non-HHNI scheme. The most energy-efficient range is that of a one-hop transmission (60 m). Figure 10 shows the results after 30 reports from 10 000 nodes. Both figures show that HHN initiation of broadcast reduces the total energy consumption of the network, however the effect becomes even more apparent for larger network sizes and after several rounds of reporting as shown in Figure 10.

4.1. The Importance of Choosing the Right Transmission Range

The previous simulations showed the *general* results for transmission ranges which only reach within few hops away. However if the transmission range can be increased further, some interesting characteristics arise. To analyze the effect of higher transmission ranges on the total network energy consumed, the

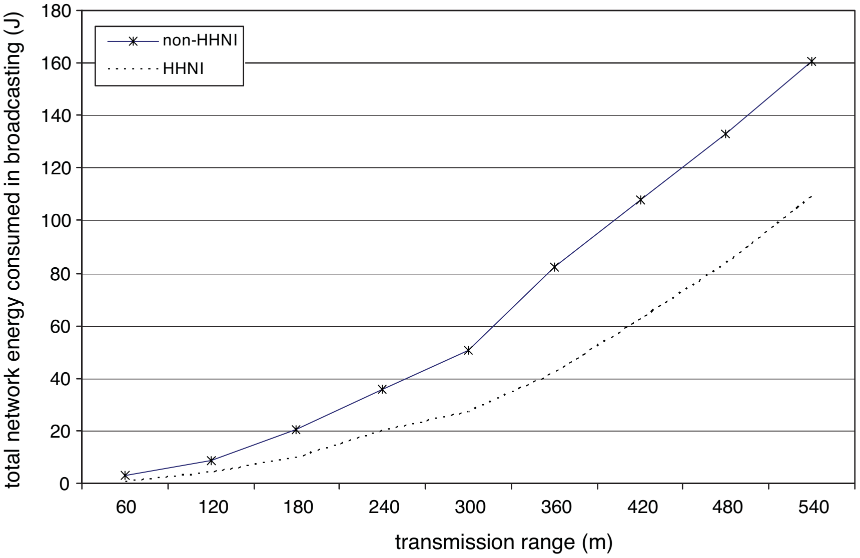


Fig. 10. Energy consumption versus transmission range for HHNI and non-HHNI routing for 10 000 nodes after 30 reports.

DOI: 10.1002/wcm

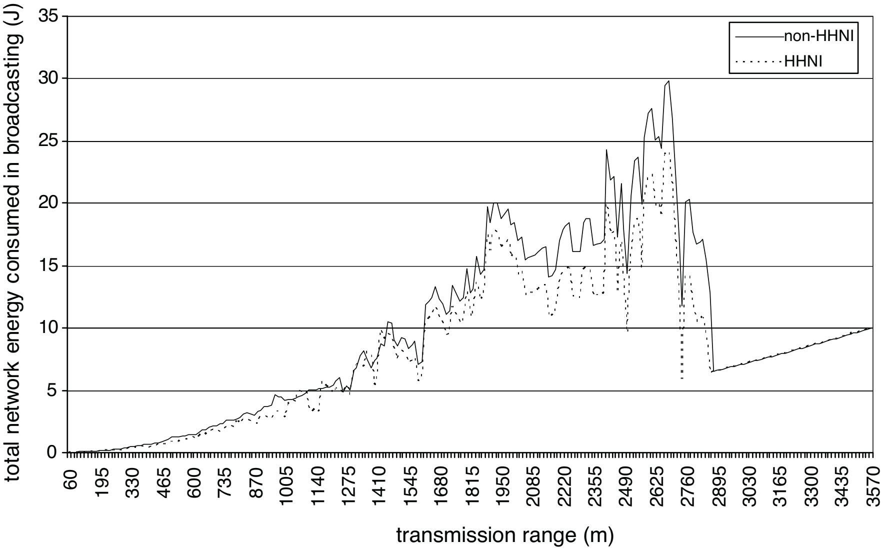


Fig. 11. Total network energy with varying transmission range.

transmission range is increased from the minimum 60 m for nearest one-hop neighbor transmission to a transmission range where the sink is able to reach all sensor nodes within the network of 2500 nodes. Figure 11 shows the results of this simulation. In the figure, the total energy consumption of the network increases with increasing range due to the extra power used by each node to transmit (proportional to the square of transmission range). However this is not linear, since the actual total energy depends on the total number of rebroadcasts of messages in the network. This is shown in Figure 12, which shows a general trend downwards as the transmission range increases. However in the figure there are rigorous fluctuations in the total number of broadcasts reflecting the strong dependence of the number of rebroadcasts on the transmission range of nodes. The network diameter in this simulation is 3535 m.

From the graph it can be seen that when nodes transmit at the range equal to the network diameter (i.e., all nodes are within one-hop range and the sink can reach all nodes within the network), the total number of broadcasts equals the number of nodes within the network. However, from both figures we

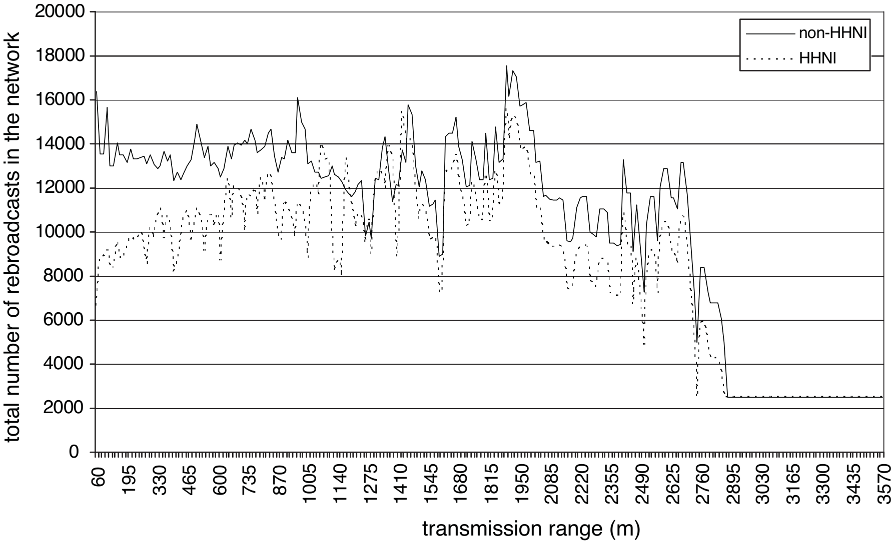


Fig. 12. Total number of rebroadcasts in the network.

DOI: 10.1002/wcm

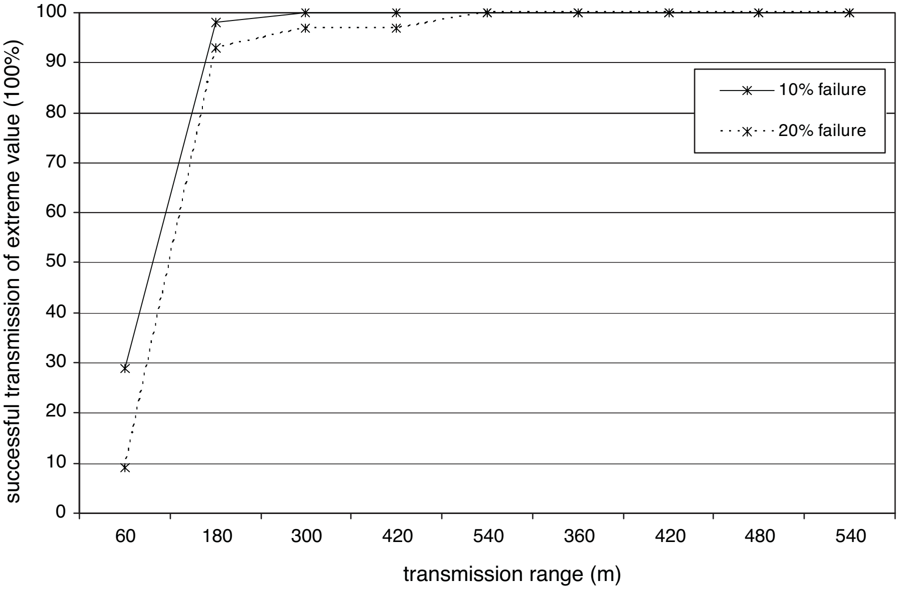


Fig. 13. Percentage of extreme value delivery in case of failed nodes.

can conclude that the use of HHN initiated routing benefits from an energy-saving point of view and for reducing the total number of broadcasts in the network. The effect of HHN initiation seems to be more dominant for lower transmission ranges. However, from an energy-saving perspective, it is best to choose low transmission ranges which reach one-hop neighbors, as the least amount of transmission power is used. Nevertheless, there is also the issue of fault tolerance, where the failure in the neighbor node can limit the network performance. The shorter the transmission range the fewer the neighbors and in the case of high node failure rate, the lower the packet delivery ratio. In our application, we are interested in forwarding the extreme sensed value of the network to the sink. Figure 13 shows the results for high node failure rates of 10 and 20% in the network. From the figure, it can be seen that having the minimum transmission range of 60 m (reaching at most 2 other neighbor nodes) is not suitable for high node failure rates, and by simply increasing the transmission range to reach up to seven sensor nodes would greatly improve the chance of delivery for even high node failure scenarios. This slight increase in redundancy of nodes does

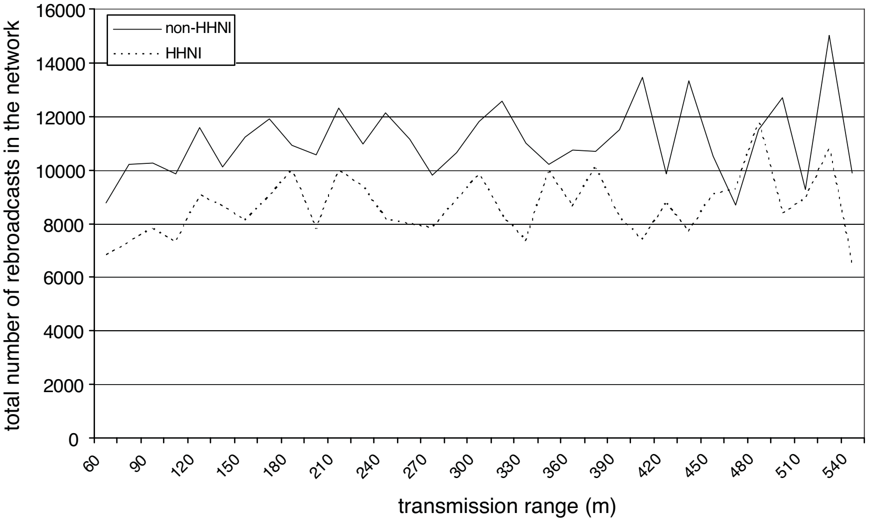


Fig. 14. Total number of rebroadcasts in the network for random topology.

DOI: 10.1002/wcm

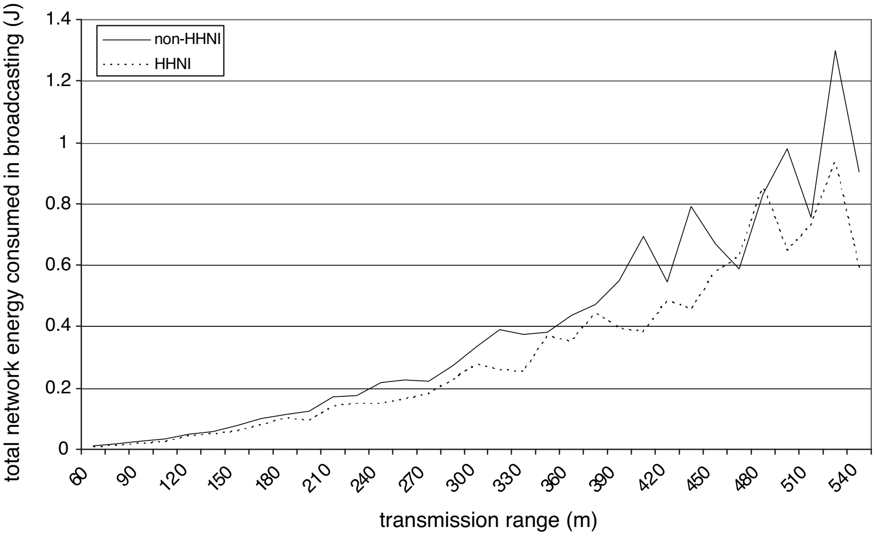


Fig. 15. Total network energy consumed in broadcasting for random topology.

not have a significant effect on energy consumption in the network and in case of high node failure scenarios the tradeoff is highly beneficial.

4.2. The Effect of Randomness on the Topology

Although the main focus of our platform was for a uniformly distributed sensor network, it is important to analyze the effect of the protocol on random topologies. Henceforth we present some simulation results for a randomly generated scenario of 2500 sensor nodes placed randomly across a square network area of 1000 m × 1000 m. The transmission range is varied. The results for the number of rebroadcasts and total network energy consumed in broadcasting is shown in Figures 14 and 15, respectively. In Figure 14, HHNI produces fewer number of broadcasts than non-HHNI routing for all the simulated transmission ranges. With regards to the amount of energy consumed, once again HHNI uses up less network energy than non-HHNI routing. Hence the results show the effectiveness of the proposed protocol even in randomly generated scenarios.

5. 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, from which their broadcasts are capable of penetrating all nodes within the network in an efficient manner. Consequently this approach is able to retrieve the extreme value of the network, be it a minimum or maximum, to the sink in an effective manner, whilst saving energy of individual sensor nodes. The principle idea may also be integrated into future routing protocols depending on the application. Simulation results show the effectiveness of the proposed approach in both uniformly distributed and randomly distributed topology networks.

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. Cheng P, Chuah CN, Liu X. Energy-aware node placement in wireless sensor networks. In *Proceedings of IEEE GLOBECOM’04*, Vol. 5, December 2004; 3210–3214.
2. Kauffman S. *At Home in the Universe: The Search for the Laws of Self-Organization and Complexity.* Oxford University Press: USA, 1996.
3. Lukachan G, Labrador M, Moreno W. Scalable and energy-efficient routing for large-scale wireless sensor networks. In *Proceedings of ICCDCS’06*, Mexico, April 2006.
4. Karp B, Kung H. Greedy perimeter stateless routing. In *Proceedings of 6th ACM MobiCom’00*, Boston, MA, August 2000.
5. Chen J, Guan Y, Pooch U. Customizing a geographical routing protocol for wireless sensor networks. In *Proceedings of ITCC’05*, Vol. 2, 4–6 April 2005; 586–591.
6. Chen K, Qin Y, Jiang F, Tang Z. A probabilistic energy-efficient routing (PEER) scheme for ad-hoc sensor networks. In *Proceedings of SECON’06*, Vol. 3, September 2006; 964–970.

DOI: 10.1002/wcm

7. Shah RC, Rabaey J. Energy aware routing for low energy ad hoc sensor networks. In *Proceedings of IEEE WCNC'02*, Orlando, FL, March 2002.
8. Yu Y, Estrin D, Govindan R. Georgraphical and energy-aware routing: a recursive data dissemination protocol for wireless sensor networks. *UCLA Computer Science Department Technichal Report*, UCLA-CSD TR-010023, May 2001.
9. Heinzelman W, Kulik J, Balakrishnan H. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of ACM MobiCom'99*, Seattle, WA, August 1999.
10. Al-Karaki JN, Kamal AE. Routing techniques in wireless sensor networks: a survery. *Wireless Communications Magazine*, December 2004.
11. Intanagonwiwat C, Govindan R, Estrin D. Directed diffusion: a scalable and robust communication paradigm for sensor network. In *Proceedings of 6th ACM MobiCom'00*, Boston, MA, August 2000.
12. Heinzelman WR. Application-specific protocol architectures for wireless networks. *Ph.D. Thesis*, Massachusetts Institute of Technology, June 2000.
13. Younis O, Fahmy S. HEED: a hybrid, energy-efficient distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing* 2004; **3**(4): 366–379.
14. Rajagopalan R, Varshney PK. Data-aggregation techniques in sensor networks: a survey. *IEEE Communication Surveys and Tutorials* 2006; **8**: 48–63.
15. Giang DV, Taleb T, Hashimoto K, Kato N, Nemoto Y. A fair and lifetime-maximum routing algorithm for wireless sensor networks. In *Proceeding IEEE Globecom'07*, Washington DC, USA, Dec. 2007.
16. Taleb T, Nait-Abdesselam F, Jamalipour A, Hashimoto K, Kato N, Nemoto Y. Design guidelines for a global and self-managed LEO satellites-based sensor network. In *Proceeding IEEE Globecom'06*, San Francisco, USA, Nov. 2006.
17. Heinzelman W, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the Hawaii Conference on System Sciences*, January 2000.

Authors' Biographies

Ehssan Sakhaee received the B.E. and Ph.D. from the University of Sydney, Australia in 2004 and 2008, respectively. He is currently a postdoctoral fellow at the Graduate School of Information Science and Technology, Osaka University, Japan. His current research interests include wireless ad-hoc and sensor networks. He is a member of IEEE.

Naoki Wakamiya received his M.E. and Ph.D. degrees from Osaka University, Japan in 1994 and 1996, respectively. He has been with Osaka University since April 1996, and he is now an Associate Professor of Graduate School of Information Science and Technology since April 2002. His research interests include overlay networks, mobile ad-hoc networks, and wireless sensor networks. He is a member of IEICE, IPSJ, ACM, and IEEE.

Masayuki Murata received the M.E. and D.E. degrees in Information and Computer Science from Osaka University, Japan, in 1984 and 1988, respectively. In April 1984, he joined Tokyo Research Laboratory, IBM Japan, as a Researcher. From September 1987 to January 1989, he was an Assistant Professor with Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. In April 1999, he became a Professor of Cybermedia Center, Osaka University, and is now with Graduate School of Information Science and Technology, Osaka University since April 2004. He has more than 400 papers of international and domestic journals and conferences. His research interests include computer communication networks, performance modeling and evaluation. He is a member of IEEE, ACM, and IEICE.

DOI: 10.1002/wcm