
Prabhudutta Mohanty *, Manas Ranjan Kabat

Dept. of Comp. Sc. & Engg., Veer Surendra Sai University of Technology, Burla, Sambalpur, Orissa, India

Received 4 November 2013; revised 16 April 2014; accepted 28 May 2014
Available online 28 June 2014

KEYWORDS
Energy efficient; Transport protocol; Reliability; Wireless Sensor Network

Abstract The two important requirements for many Wireless Sensor Networks (WSNs) are prolonged network lifetime and end-to-end reliability. The sensor nodes consume more energy during data transmission than the data sensing. In WSN, the redundant data increase the energy consumption, latency and reduce reliability during data transmission. Therefore, it is important to support energy efficient reliable data transport in WSNs. In this paper, we present a Hierarchical Energy Efficient Reliable Transport Protocol (HEERTP) for the data transmission within the WSN. This protocol maximises the network lifetime by controlling the redundant data transmission with the co-ordination of Base Station (BS). The proposed protocol also achieves end-to-end reliability using a hop-by-hop acknowledgement scheme. We evaluate the performance of the proposed protocol through simulation. The simulation results reveal that our proposed protocol achieves better performance in terms of energy efficiency, latency and reliability than the existing protocols.

1. Introduction

The Wireless Sensor Network (WSN) is a network of hundreds or thousands of tiny, resource constrained, inexpensive nodes that can sense a phenomenon, process and transmit sensed data over a wireless medium. The WSN finds its applications in various domains such as agriculture or environmental sensing, object tracking, wild life monitoring, health care, military surveillance, industrial control, home automation and security.

[1–3]. Since the WSNs are deployed in an unattended environment, the WSN applications require high reliability. The reliability of WSN is influenced by the data redundancy. The redundant data in WSN are caused either due to the slow change in phenomena or due to the same data sensed by multiple sensors. The data redundancy can be broadly classified as spatial and temporal redundancy. The spatial redundancy is caused due to multiple sensor nodes having same sensed data. The temporal redundancy is caused due to a sensor node producing same sensing value over a period. The redundant data drain the energy of the nodes, increase congestion, communication and computational overhead. The malicious nodes may take the advantage of duplicate data and cause energy drain by injecting redundant data in the network (i.e. replay attack [4]). That may lead to routing holes [5]. In WSNs, the redundant data are handled by packet sequence numbers. This technique helps the receiver to identify the duplicate data and discard it. However, a packet sequence number cannot help a
sender to control redundant data transmission. The data aggregation is another technique to eliminate redundant data. The routing schemes may use structured architecture such as cluster-based [6] or tree based [7] or structure free architecture [8,9] for data aggregation. In structured data aggregation [6,7], multiple sources send their data to the aggregation point which eliminates redundant data using various methods [10] such as statistical approaches [11,12], probabilistic approaches [13–17] and artificial intelligence [18,19]. The structure free approaches [8,9] perform dynamic data aggregation using local information so that the energy spent to build a structure can be saved. However, the energy spent due to the data transmission by the sensor nodes and data aggregation at the aggregation point cannot be avoided.

The frequency of the data sensing can be reduced to minimise the data redundancy in WSN. However, this may affect the accuracy of the data. Thus, the data should be sensed periodically and it is important to handle redundant data in WSN.

Further, the reliable data transport to the sink node must be handled by an efficient transport protocol mechanism. In this paper, we propose a framework to maximise the network lifetime and achieve end-to-end reliability by controlling the redundant data transmission with the co-ordination of BS. Our proposed framework works in two folds. First it constructs a hierarchical cluster of sensor nodes. Each cluster has a cluster head (CH) which receives the data from all the members of the cluster, aggregates the similar data and forwards it to the next level CH. This clustering technique handles the spatial redundancy. Secondly, the temporal redundancy is handled by not transmitting the temporal redundant data to the CH. The BS uses a time-out mechanism to identify the redundant data at its own side. It uses both implicit and explicit acknowledgement schemes to achieve end-to-end reliability for all the data. We propose an algorithm for BS that computes and generates an acknowledgement for each data even for redundant data without being received.

The paper is organised as follows. Section 2 presents the works done on reliable data transport over WSN. The proposed hierarchical energy efficient reliable transport control protocol is presented in Section 3. The simulation results and analysis is presented in Section 4. The summary of conclusion is presented in Section 5.

2. Related work

Many reliable transport protocols [20] have been proposed for reliable data transmission in WSNs. These are Reliable Multi-Segment Transport (RMST) [21], Event to Sink Reliable Transport (ESRT) [22], Asymmetric Reliable Transport (ART) [23], Rate-controlled Reliable Transport protocol (RCRT) [24], Flush [25], Energy-efficient and Reliable Transport Protocol (ERTP) [26], Pump Slowly Fetch Quickly (PSFQ) [27], Improved PSFQ [28] and Data-Reliable Energy Efficient Transport Layer Protocol (DREET) [29] and Distributed Caching for Sensor Network (DTSN) [30,31]. These transport protocols are analysed on the basis of reliability and energy efficiency. It is observed that protocols such as ESRT, RMST, ART, RCRT, PSFQ, Improved PSFQ and DTSN are not energy efficient.

The RMST [21] is a NACK-based upstream protocol (sensors to sink), which employs primarily timer-driven loss detection and repair mechanisms. It supports reliability with hop-by-hop recovery scheme. It introduces two modes of operation that is caching mode and non-caching mode. In caching mode, the sink node and all intermediate nodes cache the data segments and check the cache periodically for missing segments. When a node detects missing segments, it generates a NACK message which travels back to the source along the established path. In non-caching mode, the source and the sink maintain the cache and the base station monitors the integrity of the RMST data segment of the received fragments. The RMST is only suitable for reliable delivery of large blocks of data consisting of multiple segments such as JPEG image that is fragmented at the source and reassembled at the base station.

The ESRT [22] aims to provide both upstream event reliability and congestion control with minimum energy consumption. It can also reliably deliver multiple concurrent events to the base station. The ESRT guarantees only the end-to-end reliable delivery of individual event, not individual packet from each sensor node. It measures reliability by the number of packets carrying information about a particular event that are delivered to the sink. The ESRT configures the reporting frequency to achieve the desired event detection accuracy with minimum energy consumption. The ESRT always regulates the reporting frequency of all sources regardless of the congestion region. It neither prevents all losses nor retransmits lost packets. The ESRT assumes that the sink is one-hop away from all the sensor nodes, which might not be applicable to real environments.

The ART [23] is designed to provide bidirectional reliability i.e. both upstream (sensor to sink) end-to-end reliability and downstream (sink to sensor) query reliability. It also provides upstream congestion control mechanism in a decentralized way and regulates the data flow of intermediate nodes in an efficient way. A subset of sensor nodes are selected on the basis of their residual energy as essential nodes (E-nodes) to cover the domain that are required to be sensed in an energy efficient manner. A light weight ACK mechanism is adopted to guarantee reliability between E-node and sink. If ACK is not received from the sink by the E-nodes within the particular time period then the E-nodes assume congestion in the network. The E-nodes regulate the flow of the data by restraining its neighbouring non-E-nodes from sending data until the congestion is cleared.

The RCRT [24] is an upstream multipoint-to-point reliable transport protocol, which includes congestion control and explicit rate adaptation functions. The RCRT ensures reliability by using explicit end-to-end loss recovery. It implements NACK-based retransmission mechanism for end to end loss recovery, where each node along the path cache packets to support on demand loss recovery. The sink centrally performs congestion detection, recovery and rate adaptation operation. The RCRT provides end-to-end reliability of all data transmitted by each sensor to a sink. However, the RCRT reliability depends on the MAC layer retransmission which is not efficient. A single packet loss may force rate reduction as the congestion detection depends on loss recovery time. The RCRT does not address the issue of contention.

The Flush [25] is a reliable high good put bulk data transport protocol that provides end-to-end reliability. In Flush, the sink schedules the data transfer for each node in a round robin fashion to support single data flow and to avoid inter-path interference. To improve channel utilisation, the rate
allocation algorithm follows two basic rules. Firstly, a node is allowed transmitting only when its successor is free from interference. Secondly, sending rate of a node must be less than the sending rate of its successor. The Flush uses a queuing technique to buffer packets during transient rate mismatches, which are typically due to changes in link quality. However, the rate plays a vital role in Flush that is if the rate is too high then it causes self-interference while too low rate will cause poor capacity utilisation.

The ERTP [26] is an upstream energy efficient transport protocol designed to provide end-to-end statistical reliability for low data streaming WSN applications. The reliability of ERTP is determined by the quantity of data packets received at the sink rather than the reliability of each data packet. It achieves end-to-end reliability by controlling the reliability at each hop and maximises the energy efficiency using stop and wait hop-by-hop Implicit Acknowledgement (IACK) for loss recovery. However, The ERTP assumes low data rate (negligible collision), overheard packet transmission cost is low for single hop neighbour and collisions due to transmission from two neighbour nodes at same time are negligible.

The PSFQ [27] distributes data hop-by-hop from sink to sensors (downstream) to meet the unique resource challenges of WSNs with a focus on reliability, scalability and robustness. The PSFQ contains three protocol functions i.e. message transmission (pump operation), local loss recovery (fetch operation), and selective status reporting (report operation). In the pump operation sink slowly injects packets into the network until all the data segments have been sent out. The reliability in PSFQ is achieved with a negative acknowledgement (NACK) based on quick fetch operation. A sensor node can go into fetch mode once a loss is detected using gap sequence. The report operation is designed to feedback data delivery status to users in a simple and scalable hop by hop manner. Each node along the path towards the source node will piggyback their report message by adding its own status information into the report and then propagate the aggregate report towards the user node. To avoid looping, each node ignores its own report, if its own ID is found in the report. However, the PSFQ strictly follows in sequence forwarding of data packets due to the rate plays a vital role in Flush that is if the rate is too high then it causes self-interference while too low rate will cause poor capacity utilisation.

The PSFQ [27] distributes data hop-by-hop from sink to sensors (downstream) to meet the unique resource challenges of WSNs with a focus on reliability, scalability and robustness. The PSFQ contains three protocol functions i.e. message transmission (pump operation), local loss recovery (fetch operation), and selective status reporting (report operation). In the pump operation sink slowly injects packets into the network until all the data segments have been sent out. The reliability in PSFQ is achieved with a negative acknowledgement (NACK) based on quick fetch operation. A sensor node can go into fetch mode once a loss is detected using gap sequence. The report operation is designed to feedback data delivery status to users in a simple and scalable hop by hop manner. Each node along the path towards the source node will piggyback their report message by adding its own status information into the report and then propagate the aggregate report towards the user node. To avoid looping, each node ignores its own report, if its own ID is found in the report. However, the PSFQ strictly follows in sequence forwarding of data packets due to which other nodes remain idle until they get data packets in sequence. This increases the latency of the protocol. The PSFQ is unable to recover continuous lost packets when wireless channel error rate is high. This degrades the reliability in PSFQ due to in sequence forwarding.

The improved PSFQ [28] is proposed to overcome the shortfalls of PSFQ [27] caused due to sequence forwarding and pump and repair operations. The improved PSFQ introduces two schemes i.e. out of order forwarding and rescheduling of pump and repair packets to overcome the lacunae of PSFQ. The sensor node performs a duplicity check for received data packets. All duplicate data packets are discarded where as new data (non-duplicate) packets are cached and schedule for transmission. The improved PSFQ cached data in an array called “buffer” to store and assign sequence number for the non-duplicate data. The PSFQ minimises the unnecessary delay caused due to in sequence forwarding of data packets. It also solves the problem of NACK implosion which arises in PSFQ.

The DREET [29] selects a subset of sensor data as key data and rest data as ordinary data. The first sensor data of each event and any data that are greater than or less than the predefined value are also called key data. The DREET provides reliable transport for key data rather than all data. In case of congestion, the key data packets are retransmitted to achieve reliability whereas the ordinary packets are dropped from the buffer of intermediate nodes. The lost ordinary data packets are computed using relative standard deviation between sensor data and sink data. The key data packets are acknowledged by the sink node to ensure high reliability whereas acknowledgements are ignored for ordinary data to conserve energy. However, it drops ordinary data packet with increase in congestion and provides no reliability to theses packets.

The DTNS [30] and its variant [31] improve reliability by employing cache at selected intermediate nodes. These nodes can store and retransmit packets that are lost during transmission. The DTNS is designed to provide different grades of reliability such as full or differentiated reliability services. The full reliability mode aims to deliver all the packets at the sink. On the other hand the differential reliability does not guarantee delivery of all packets to the sink. The full reliability level is employed with the help of a Selective Repeat Automatic Repeat request (SR-ARQ) using negative acknowledgement (NACK) and positive acknowledgement packet (ACK) for loss recovery and packet control. Both NACK and ACK are to be sent by the receiver upon request by the sender using Explicit Acknowledgement Request (EAR) which can be piggybacked in the data packets. The ACK/NACK mechanism results in high message overhead which affects the overall energy efficiency. In the improved DTNS [31], the caching decisions are taken independently based on the criteria such as hop count, link quality and energy availability. The impact of different cache partitioning policies on the transmission cost is evaluated. However, the caching at intermediate nodes requires large storage which may not be suitable for resource constrained WSNs and an efficient cache management mechanism is also required to minimise the unnecessary cache occupancy.

3. Hierarchical Energy Efficient Reliable Transport Protocol (HEERTP)

In this section, we present the hierarchical cluster formation followed by intelligent data transport for redundant data and missing redundant data computation at base station. Finally, we present acknowledgement scheme to achieve reliability with various case studies.

The HEERTP is a hierarchical cluster based transport protocol that minimises energy consumption by reducing redundant data transport over WSN. In the proposed protocol, the sensor nodes minimise the transmission of sensed redundant data with the co-ordination of the BS. Our proposed framework creates a hierarchy of clusters consisting of sensors for data gathering within the network. One node in the cluster is designated as the cluster head (CH). The CH is responsible for collecting data in the group and forwards the collected data towards the Base Station (BS) directly or through hierarchy of CHs. The CH is selected on the basis of co-ordinate position and residual energy of the nodes. Many methods are proposed in the literature to select a CH, leader or root [6,32]. The cluster formation consumes a significant amount of energy in WSN. The proposed protocol adapts a simple cluster formation technique to avoid the computational complexity.
3.1. Hierarchical cluster formation

The HEERTP form clusters of different levels with the support of BS. The leaf nodes in the cluster hierarchy forwards sensed information to their CH. The CH forwards the gathered data to the next level CH and finally the CH nearest to the BS, forwards the information to BS. We call the CH nearest to the BS and top in the hierarchy as root level cluster head (RCH). To reduce the cluster formation overhead, the BS is allowed to select a RCH. The level of cluster formation is controlled by a parameter called maximum hop count value (Mhc), which is also computed by the BS. The Mhc is computed by the BS using Eq. (1).

\[
M_{hc} = \frac{D}{d} - 1
\]

where \( D \) is the maximum possible distance between two ends of a sensor field and \( d \) is the transmission range of a sensor node as shown in Fig. 1. If the sensor field area can be enclosed within a rectangular area then \( D \) is the diagonal of the rectangle and computed as \( \sqrt{a^2 + b^2} \), where \( a \) and \( b \) are the two sides of the rectangle that encloses entire sensing field. In case of a circular sensor field, \( D \) can be considered to be the diameter of the circle that encloses sensing field.

The main idea of level wise cluster formation is diagrammatically represented in Fig. 2. The sensor nodes are deployed randomly in the sensor field. We assume that a sensor node is aware of the co-ordinate position of its own and as well as the BS at the time of node deployment. The nodes that are interested to be the RCH, compute their distances to the BS. If the computed distance (\( d_{node} \)) is less than the transmission range of a sensor node (\( t_i \)) then it sends a request message to BS as shown in step 1 of Fig. 2. The request message contains sensor node’s ID (SNID), remaining battery power (Resbattpow), distance and base station ID (BSID). The BS calculates the ratio of remaining battery power to the distance (\( \rho \)) of each requesting node. The node that has the maximum value of \( \rho \) is selected as RCH. The base station replies back to the selected RCH with an acknowledgement as shown in step 2 of Fig. 2. The acknowledgement (ACK) packet contains BSID, SNID and a hop count value. After receiving ACK packet, the RCH sets it hop count value one less than the received hop count value. The RCH node broadcasts the advertisement message to the neighbour nodes which are within the radio range of RCH to form a cluster as shown in step 3 of Fig. 2. The advertisement message contains cluster head ID(CHID), SNID, Position and hop count. In root level cluster formation phase the CHID is the ID of root level cluster head. The advertisement messages send by the CH or RCH sets hop count one less than its own hop count value. Nodes within the neighbouring area of RCH decide to join the cluster send a joining message to RCH as shown in step 4 of Fig. 2. The cluster formed with this RCH is known as root level cluster, first order cluster or one hop cluster. Once root level cluster formation is over, non-cluster head nodes broadcast CH advertisement message again to form next level clusters as shown in step 5 of Fig. 2. The advertisement message contains same fields as described earlier in RCH. It may happen that a node will receive more than one cluster advertisement message. If a node receives advertisement message from multiple CH then it selects a CH to join, on the basis of maximum hop count and minimum distance. Node decides to join a cluster whose hop count value is highest among received advertisement message since the higher the hop count value is, the smaller the distance from the BS. If the node receives more than one cluster advertisement message with same hop count value then the node randomly selects a node among the received advertisement message. The step 6 of Fig. 2 shows that nodes send joining message to form second order cluster or two hop cluster. This process continues till all the nodes in the network are grouped into the clusters and forms the cluster hierarchy. The steps 7 and 8 of Fig. 2 shows the hierarchy of cluster formation. If a node could not receive any cluster advertisement message after a predefined time period, it tries to find the nearest cluster and associates with it by increasing its radio range step by step. Fig. 3 shows the algorithm involved in cluster formation phase. The BS initiates the re-cluster formation by broadcasting a control packet. The re-clustering is initiated by the BS when it receives a control message from the RCH. The control message is sent to BS when \( Resbat_{pos} \leq 2 \times T_{energy} \). The \( T_{energy} \) is threshold energy of a node that is computed using Eq. (2).

\[
T_{energy} = \min(S_{energy}) + T_{req\_energy}
\]

where \( S_{energy} \) is energy required to sense the data and \( T_{req\_energy} \) is the energy required to transmit the data packet to the destination. The \( T_{req\_energy} \) is computed using Eq. (3).

![Figure 1](image_url) Estimating maximum necessary hop-count in the sensor field.
Figure 2  Cluster formation in a sensor field.
A fixed dissipating energy ($E_{\text{elec}}$) is spent in transmitting and receiving a packet. The transmission distance $t_d$ affects the power consumption when sensor nodes want to transmit data. An extra cost proportional to $t_d^2$ is spent for amplifier ($E_{\text{amp}}$) in transmitting a packet. Table 1 contains symbols used in the equations.

### 3.2. Intelligent data transport

The data transmission consumes comparatively more energy than sensing and processing. The proposed protocol HEERTP, intelligently decides to transport data to conserve more energy. The data sensing phase begins after cluster setup phase followed by data transmission phase. The generic structure of the data transmission of the proposed HEERTP model is shown in Fig. 4. The function of the proposed transport protocol is divided into three parts i.e. at the sender side (sensor node), $CH$ and the destination side (BS). The HEERTP works in four phases at the source end. It starts with sensing phase followed by data validation phase, transmission phase and ends with acknowledgement verification phase.

Each sensor node senses the phenomena and gathers information in data sensing phase. A sensor node initiates the data transmission phase to send the sensor data to its $CH$. The generic structure of the data transmission of the proposed HEERTP model is shown in Fig. 4. The function of the proposed transport protocol is divided into three parts i.e. at the sender side (sensor node), $CH$ and the destination side (BS). The HEERTP works in four phases at the source end. It starts with sensing phase followed by data validation phase, transmission phase and ends with acknowledgement verification phase.

$$T_{\text{req\ energy}} = E_{\text{elec}} + E_{\text{amp}} \times t_d^2$$ (3)

Algorithm Cluster Formation()
Executed at the sensor node
begin:
  interested nodes $N_i$ to be a $R_{CH}$ computes $d_{\text{NIS}}$
  if($d_{\text{NIS}} < t_r$) then
    transmit{request message};
  else
    wait;
  endif;
  if{receive (ACK $CH$)= true} then
    set hop count = $M_{\text{hc}}$ - 1;
    transmit{advertisement message};
    if{receive{join message} = true} then
      store $SN_{i\alpha}$ and co-ordinate;
    endif;
  endif;
  if{(member (cluster)=true) and (!CH)} then
    set hop count=hop count-1;
    transmit{advertisement message};
  else
    if{(member (cluster)=false) and (!CH) and
    (receive{advertisement message}=false) and
    (time out()=true)) then
      find a $CH$ and join it;
    else
      if{(member (cluster)=false) and (!CH) and
      (receive{advertisement message}=true) and
      (time out()=true)) then
        select the advertised node as $CH$ and join it;
      endif;
    endif;
  endif;
end;
Algorithm Cluster Formation BS()
Executed at the Base station
begin:
  initialization $T_{\text{energy}} = \min(S_{\text{energy}}+T_{\text{req\ energy}}, d = t_r$;
  compute $D$ from sensing field;
  compute $M_{\text{hc}} = (D/d) - 1$;
  select $RCH$();
  set hop count = $M_{\text{hc}}$
  transmit (ACK $CH$);
  if{$Res_{\text{batt\ pow}} \leq 2* T_{\text{energy}}$} then
    initiate cluster formation();
  endif;
end;
select $RCH()$
begin:
  for i=1 to N
    compute ratio ($\rho$) as $Res_{\text{batt\ pow}}$ to $d_{\text{NIS}}$
  endfor;
  select NodeI with max ($\rho$) as $R_{CH}$;
end;

Figure 3 Pseudocode for hierarchical cluster formation.
sensing phase when a sensor node has it is \( \text{Resbatt\_pow} \geq \text{Thenergy} \) (Table 1), the \( \text{Thenergy} \) is computed using Eq. (2). These gathered data are passed to the validation phase. The data validation phase smartly decides the data forwarding to BS. A valid data is transmitted to BS and an invalid data is forbidden to transmit. The validity of the current sensed data is decided by comparing it with the previous sensed data. A data is valid for transmission if it is non-redundant or its number of redundancy is greater than the predefined threshold value (\( \text{th} \)). The \( \text{th} \) value is maximum number of times a redundant data can be ignored from transmission by a sensor node. The \( \text{th} \) is set and decided by the user based on the application requirements. If the \( \text{th} \) value is equal to 0 then it is same as transmitting all the data packets including redundant data. If the \( \text{th} \) value is set as one then it suppress redundant data transmission once which is almost produces equal result as \( \text{th} = 0 \). So the \( \text{th} \) value is selected at least greater than two for the applications that producing much less redundant data. The greater the \( \text{th} \) value increases energy efficiency by reducing number of redundant data transmission but the \( \text{th} \) should not be too high so that communication is mislead and data accuracy is lost. At the time of node deployment, the \( \text{th} \) value is required to be set manually by the user and it can be adjusted by broadcasting a control message through BS if required to suppress redundant data transmission from the sensing field. In our proposed protocol we assume the value of \( \text{th} \) as five, which is selected incrementing the value one by one during simulation. The transmission phase is triggered by valid data forbidding the invalid data from transmission. In the validation phase, the sensor node buffers the sensed data till an acknowledgement is received for the sensed and transmitted data. The data validation phase assigns packet sequence number (\( pseq\ no \)) to the data. The \( pseq\ no \) of a non-redundant is set to zero. Then, the \( pseq\ no \) is incremented by one for every sensed redundant data. This packet sequence number (\( pseq\ no \)) increment process continues till the number of redundant data sensed reaches the to the value \( \text{th} \). When the same data is sensed \( \text{th} \) times the packet \( pseq\ no \) of the next packet is set to zero. The \( pseq\ no \) and acknowledgement number (\( ACK\ No \)) computation for the received data is illustrated in Table 2.

Let \( D_1 \) and \( D_2 \) are two different data values sensed by the sensor node. Suppose \( D_2 \) value is sensed repeatedly by the sensor node then the \( pseq\ no \) for \( D_2 \) is computed as \( pseq\ no = 0 \).

### Table 1: Notation table.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_{CH} )</td>
<td>Root level cluster head</td>
</tr>
<tr>
<td>( CH )</td>
<td>Cluster head</td>
</tr>
<tr>
<td>( d_{NiBS} )</td>
<td>Distance between a node (( N_i )) and base station</td>
</tr>
<tr>
<td>( t_r )</td>
<td>Transmission range of a sensor node</td>
</tr>
<tr>
<td>( SNID )</td>
<td>Sensor node ID</td>
</tr>
<tr>
<td>( BSID )</td>
<td>Base station ID</td>
</tr>
<tr>
<td>( \text{Resbatt_pow} )</td>
<td>Remaining battery power of a sensor node</td>
</tr>
<tr>
<td>( \rho )</td>
<td>Ratio between ( \text{Resbatt_pow} ) and ( d_{NiBS} )</td>
</tr>
<tr>
<td>( M_{hc} )</td>
<td>Maximum hop count to form cluster</td>
</tr>
<tr>
<td>( C_ID )</td>
<td>Cluster head ID</td>
</tr>
<tr>
<td>( \text{Thenergy} )</td>
<td>Threshold energy required for one round communication</td>
</tr>
<tr>
<td>( pseq\ no )</td>
<td>Packet sequence number</td>
</tr>
<tr>
<td>( \text{th} )</td>
<td>Threshold value set by the user for number of redundant data</td>
</tr>
<tr>
<td>( ACK\ NO )</td>
<td>Acknowledgement number</td>
</tr>
<tr>
<td>( S_{energy} )</td>
<td>Energy required to sense a phenomena</td>
</tr>
<tr>
<td>( T_{req\ energy} )</td>
<td>Energy required to transmit a data packet</td>
</tr>
<tr>
<td>( E_{elec} )</td>
<td>Energy spent in transmitting and receiving a packet</td>
</tr>
<tr>
<td>( E_{amp} )</td>
<td>Energy spent in amplification</td>
</tr>
</tbody>
</table>

### Table 2: Packet sequence number and Acknowledgement number computation.

<table>
<thead>
<tr>
<th>Sense data</th>
<th>( pseq\ no/ACK\ NO )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( D_1 )</td>
<td>0</td>
</tr>
<tr>
<td>( D_2 )</td>
<td>1</td>
</tr>
<tr>
<td>( D_2 )</td>
<td>2</td>
</tr>
<tr>
<td>( D_2 )</td>
<td>3</td>
</tr>
<tr>
<td>( D_2 )</td>
<td>4</td>
</tr>
<tr>
<td>( D_2 )</td>
<td>5</td>
</tr>
<tr>
<td>( D_2 )</td>
<td>0</td>
</tr>
</tbody>
</table>
no = pseq no + 1 up to th times. Let us assume that the value of th is five. If D2 is sensed after th times then pseq no is set to zero. When pseq no is zero the data is valid for transmission. The valid data with packet sequence number transmitted to the BS. In transmission phase, the transmitter is activated first and then distance to the destination node is adjusted. The transmission phase forwards the data packet towards the BS through CH. The pseudocode of the functionalities of the proposed protocol at the sender side is presented in Fig. 5.

The CH forwards the received data from the sensor node to BS. The functionalities of CH is divided into two phases i.e. data transmission phase and acknowledgment analysis phase. In data transmission phase, the transmitter is activated and the transmission distance is adjusted to the destination node (either to the next level CH or BS if it is a RCH). In acknowledgment analysis phase, the sender analyses the received acknowledgement from the destination. In this phase, the acknowledgement number is cross-verified with the packet sequence number of the sent data.

The pseudocode of the HEERTP functionalities at the destination side is shown in Fig. 6. It is divided into two phases i.e. data update phase and acknowledgement generation phase. The update phase for a node is initiated either after receiving data from source node or after the data receive period timeout occurs. The update phase is carried out by the BS. The BS maintains a data update table which is shown in Table 3. The data update table contains sensor node id, node status, node co-ordinate values, time at which previous data is received, a counter value that counts the number of redundancy, previous data value received for the node, current data value and the packet sequence number. When a non-redundant data is received at the BS then the current data is replaced by the received data value, sets the previous data received time with current time and updates the previous data value field. The packet sequence number and counter are reset to zero. If time out occurs for a sensor node for receiving a data then BS treats it as redundant data. For redundant data BS checks the counter if the counter is less than th value, the current data is replaced by the previous data value, sets the previous data received time with current time and the packet sequence number and counter value are incremented by one. When a redundant data is received at the BS and the counter value is equal to th then the previous data received time is replaced by the current time and the packet sequence number and counter value are reset to zero. If counter value is equal to th and time out occurs then BS checks the status of the node by sending a control packet. When BS assures that the node is dead then the status of the node is updated from alive (A) to dead (D) and no further update is carried out for the node.

The entire update process is based on the co-ordination between a sensor node and BS. Sensor node forwards data to the cluster head. Cluster heads forward the data packet to upper level CH in the hierarchy and finally RCH forwards the data to the BS. When a cluster head forwards the data towards the upper level cluster, the sensor node within the cluster overhears the transmission and ensures about their data forwarding towards the BS. This is an implicit acknowledgement (IACK) for sensor nodes. The implicit acknowledgements are received when data is forwarded level by level up to root level clusters. The BS sends an explicit acknowledgement (EACK) to RCH after receiving sensed data, as there is no further data forwarding at the BS. Fig. 7 demonstrates the implicit and explicit acknowledgement. In acknowledgement generation phase, the BS analyses the received data packet from the sensor node. For a non-redundant data packet received after the predefined threshold value (i.e. th), the packet sequence

Algorithm Data_Gathering( Resart_pow )
begin:
    initialize Th_energy = S_energy + T_req_energy, pseq_no=0;
    initialize prevdata=!, read data=!, node_th_count=0, th;
    while(Resart_pow >= Th_energy) do
        read_data = sense_data();
        if(valid_data(read_data)=true)
            transmit_data(read_data);
        else
            set node_th_count= node_th_count +1;
            set pseq_no= pseq_no +1;
        endif;
    end while;
    if (ACK_NO!=pseq_no) then
        transmit_data(read_data);
    endif;
end;
valid_data(read_data)
begin:
    if((prevdata = 0)or(read_data = prevdata) or
       (node_th_count=th)) then
        set prevdata= read_data;
        set pseq_no=0;
        node_th_count=0;
        return true;
    else
        return false;
    endif;
end;

Figure 5 Pseudocodes of the functionalities at sender side.
number (pseq no) is checked. If the packet sequence number is zero then ACK 0 is generated. For redundant data ACK is generated with a pseq no = previous pseq no + 1. A negative acknowledgement (NACK) is generated for the data packet, after ignoring redundant data for th times within timeout period. When a redundant data packet is received and the packet sequence number is not equal to zero then data is discarded. It assumes that the data with packet sequence number not equal to zero is sent by the source node due to loss of sent acknowledgement. Therefore, the previous ACK is retransmitted.

3.3. Reliability and case study

One of the most important phase, that ensures the reliability in the sensor side is acknowledgement verification phase. In this phase, the acknowledgement number is cross-verified with packet sequence No. If both the numbers match then the node confirms that transmission is successful. Otherwise, the sensed data is retransmitted. If any packet loss occurs in BS, then data is retransmitted from the CHs hierarchically. If CHs fail to retransmit the data then it is recovered from the source node. Here the protocol performs intermediate caching like DTSN [30,31]. The CH caches the data packets and retransmits them on demand. This phase helps to recover from the problem of packet loss and achieve end-to-end reliability with minimum overhead. The CH reduces traffic overhead by retransmitting lost packets and suppressing NACKs. We consider certain scenarios to understand the loss recovery and maintain reliability assuming the value of th to be five.

Table 3  Data update table for base station.

<table>
<thead>
<tr>
<th>Node id</th>
<th>Node status</th>
<th>Node co-ordinate points</th>
<th>Previous data received time</th>
<th>Counter</th>
<th>Previous data</th>
<th>Current data</th>
<th>Packet sequence. no</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>A</td>
<td>50.80</td>
<td>10:15</td>
<td>3</td>
<td>40</td>
<td>40</td>
<td>3</td>
</tr>
<tr>
<td>N2</td>
<td>A</td>
<td>75.94</td>
<td>10:16</td>
<td>0</td>
<td>42</td>
<td>43</td>
<td>0</td>
</tr>
<tr>
<td>N3</td>
<td>D</td>
<td>30.43</td>
<td>9:45</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Figure 7  Implicit and explicit acknowledgement scheme.
downstream CH or a sender receives an implicit acknowledgement (IACK). The BS acknowledges to the $R_{CH}$ with an explicit acknowledgement after receiving the data. Fig. 8 shows the acknowledgement operations for normal flow. The $pseq$ no of the redundant packets are numbered as 1, 2, ..., 5 and the EACKs are numbered as 1, 2, ..., 5 accordingly. The $pseq$ no of the 6th redundant data packet is reset to 0 and the packet is transmitted as a new data. Table 2 shows the EACK numbers for normal flow.

**Case-2:** Impact of redundant packet loss from source to CH and CH to BS after $th$ time over: In this case, we consider a scenario where redundant data packet is lost after threshold time ($th$). The 6th redundant data is transmitted to the CH with a $pseq$ no 0 after ignoring redundant data for five times. Fig. 9 shows the redundant data packet loss after $th$ times. If the 6th redundant data packet is lost during the transmission from the sender to CH or one CH to another CH then the IACK of that packet is not received. Thus the sender or CH retransmits the data to the upstream CH after the timeout. However, if the data is lost within $R_{CH}$ and BS then the BS sends NACK to $R_{CH}$. The $R_{CH}$ retransmits the data to BS from its cache.

**Case-3:** Impact of loss of ACK for redundant data: Fig. 10 shows the impact of ACK loss for redundant data. When the sensor senses a redundant data it does not transmit to the BS. All the CHs in the path from sender to BS automatically assume the redundant data after the timeout and keep it in their cache. The BS transmits the EACK to the sender for the redundant data. If the EACK is lost in the downstream link ($CH_k, CH_{k-1}$) then $CH_{k-1}$ sends a control packet with any random non-zero $pseq$ No. then the $CH_k$ retransmits the EACK received from the BS.

**Case-4:** Impact of packet loss for non-redundant data: Fig. 11 shows the impact of loss of non-redundant data during transmission. If the data is lost within the sender and CH then the sender retransmits the packet after timeout of IACK. Similarly, if the packet is lost at the BS then the BS transmits an EACK with a different sequence number assuming the data as a redundant data. The $R_{CH}$ discards the EACK and retransmits the non-redundant data packet.

### 4. Performance evaluation

In this section, we present the performance evaluation of our proposed protocol through simulation in NS-2.29 [33]. We compare the simulation results of our proposed protocol HEERTP, with other existing protocols like ERTP [26], Improved PSFQ [28], DREET [29], cluster with direct transmission [34], Structured data aggregation i.e. Hybrid Energy Efficient Distributed clustering (HEED) [6] and structure free data aggregation i.e. RAG [9] and the cached based intermediate loss recovery protocols (DSTN) [30,31]. The simulation parameters of our proposed protocol is described in Table 4. We run the simulation with parameters like different data rate and an average data redundancy of 20% and 40% respectively. We assume that only same type of data packets could be aggregated in the CHs along the way to the BS. The aggregated packets are identified by adding an extra field containing information about the number of data aggregated to the data packets. It is also assumed that a maximum of 65 numbers of data packets can be aggregated during data transmission. The packet generation rate increased step by step from 1 to 100 pps (packets/sec). The proposed protocol is analysed in...
Figure 9  Impact of redundant packet lost after threshold time over.

Figure 10  Impact of loss of ACK for redundant data.
terms of average energy consumption, miss ratio and average data delivery ratio through the simulation. The miss ratio is computed as the ratio of packets lost due to buffer overflow at the intermediate CHs to the number of packets transmitted to the BS. We randomly distribute 100 nodes in a sensor field of area \(100 \times 100\) m\(^2\) with sink located at location \((25, 50)\). The initial energy level of each sensor node is assumed to be 1.0 J. The data packet length is assumed to be of 30 bytes. The radio channel is assumed to be symmetric. In order to construct the best possible result we run the simulation hundred times up to 3600 ms and consider the average of that as output.

![Figure 11](image1.png)

**Figure 11** Impact of packet loss for non-redundant data.

Fig. 12 shows the energy level vs number of communications. We find the number of nodes \((n_c)\) whose energy level becomes less than minimum threshold after a specific time \(\tau\) (3600 ms). Then we also compute the total number of data sensed and transmitted/not transmitted by those nodes \((d_c)\). The average number of communications is computed as the ratio of \(d_c\) and \(n_c\). It is observed from Fig. 12 that the network can last long if HEERTP is used for data transport than E RTP, Improved PSFQ, DREET and DTSN. If there is no

![Figure 12](image2.png)

**Figure 12** Energy level vs Average number of communication of different transport protocols.
redundant data then HEERTP performs at par ERTP. How-
ever, it is not the practical scenario. The HEERTP outper-
forms the existing protocols if there is a 20% redundant data
in the network. Fig. 12 shows the performance of HEERTP
with 20% and 40% redundant data respectively. It is observed
that HEERTP performs better than the existing protocols.
This happens due to the effect of the data redundancy. The
ERTP and Improved PSFQ transmit all sensed data without
any consideration of redundancy. The DREET considers reli-
bility for a subset of data (key data) but it transmits all sensed
data without redundancy check. The DREET only avoids
retransmission of ordinary data to conserve energy whereas
our proposed protocol conserves more energy discarding the
redundant data. Therefore, the HEERTP performs better than
the ERTP, improved PSFQ, DREET and DTSN in terms of
energy consumption when the data redundancy increases.

Fig. 13 shows the packet delivery ratio for each hundred
rounds of communication. In this simulation, we introduce
artificial packet loss in the network to analyse the percentage
of packets delivery ratio in the network. The percentage of
packet delivered is calculated for each 100 data packets
received at the BS. The packet delivery ratio is computed as
the ratio of 100 to the sum of 100 and number of packets
dropped. The packets delivery ratio is computed for each
100 rounds of communication from 1 to 1000. It is observed
that HEERTP performs better than the existing protocols
because it suppresses the redundant data and minimises the
loss occurred due to buffer overflow.

Figs. 14 and 15 show the average energy consumption and
data miss ratio vs data rate respectively. As the data rate
increases, the network becomes more and more congested.
The ERTP, improved PSFQ, DREET and DTSN transmit
all the data without any redundancy check. Thus the energy
consumption increases due to the transmission of all data.
Subsequently, due to the increase in congestion, more number
of packets are dropped at the intermediate routers. Thus, the
data miss ratio increases with the increase in data rate. The
HEERTP suppresses the redundant data at the source end
which not only saves the energy but also prevents the buffer
overflow at the CHs. It can be observed from Fig. 14 that
the average energy consumption of HEERTP is less as com-
pared to ERTP, improved PSFQ, DREET and DTSN. Simi-
larly, it can also be observed from Fig. 15 that the data miss
ratio of HEERTP is less than ERTP, improved PSFQ, DREET
and DTSN.

There are several data aggregation protocols developed to
handle data redundancy in WSN. To study and compare the
performance of our proposed protocol with various data
aggregation protocols, we simulate cluster with direct trans-
mision [34], Structured data aggregation i.e. Hybrid Energy
Efficient Distributed clustering (HEED) [6] and structure-free
data aggregation i.e. RAG [9]. The energy level vs average
number of communications of direct communication with
cluster, multi-hop communication cluster, HEED, RAG and
HEERTP is shown in Fig. 16. The direct transmission with
cluster and multi-hop cluster transmits all sensed data without
considering data redundancy. Therefore, these two protocols
incur more energy consumption than HEERTP. The data
aggregation techniques aggregate the redundant data to reduce
number of transmission. However, the energy spent due to the
data transmission by the sensor nodes and data aggregation at
the aggregation point cannot be avoided. It is observed that
the performance of HEERTP increases with increase in
redundancy. The packet delivery ratio of cluster with direct
communication, multi-hop cluster, HEED, RAG and
HEERTP for each 100 rounds of communication is shown in
Fig. 17. Figs. 18 and 19 show average energy consumption
and miss ratio during different data rate. The increase in data
rate not only leads to the increase in data redundancy but also increase in congestion. The data aggregation can be better with increase in redundant data packets. However, when the number of redundant data packets increase the packet drop will increase in the aggregation point. It not only increases the packet miss ratio but also increases energy consumption at aggregation point. The proposed protocol suppresses the redundant data at source end for which the CHs are less overwhelmed with the redundant packets. Therefore, the HEERTP performs better than the existing data aggregation protocols in terms of energy consumption and data miss ratio if the data rate increases.

5. Conclusion

In this paper, we propose a framework for energy efficient reliable data transport in Wireless Sensor Networks. The proposed framework constructs a cluster based structure for handling spatial redundancy by aggregating redundant data at CHs. It also minimises the temporal redundant data transmission with the co-ordination of BS and the BS detects the redundant data even without receiving from the sensor node. The proposed method identifies the redundant data at the receiver side when timeout occurs. If the receiver receives non-redundant data then it updates data table. Each time a data is sensed, it requires one comparison for redundancy check. Therefore, the time complexity of the algorithm run at the sender side is $O(1)$. Similarly the algorithm run at the receiver side compares packet sequence number with time counter for acknowledgement generation and thus has time complexity of $O(1)$. We introduced both implicit and explicit acknowledgement to achieve end-to-end reliability. The performance of the proposed protocol is studied through simulation and it is observed that our protocol outperforms existing methods in terms of energy and packet delivery ratio.

References


Prabhudutta Mohanty is currently pursuing his Ph.D from Veer Surendra Sai University of Technology, Burla, India. He has received his M.Sc in Computer Science and Master degree in Computer Science and Technology from IASE Deemed University and Tezpur University respectively. His area of interest is Wireless Sensor Network, Internet and QoS routing. He has published 10 research papers in various international journals and conferences.

Manas Ranjan Kabat has received his M.E. degree in Information Technology and Computer Engineering from Bengal Engineering College, India, and the Ph.D degree in Computer Science and Engineering from Sambalpur University, India. He is currently working as Reader and Head of the Department of Computer Science and Engineering at Veer Surendra Sai University of Technology, Odisha, India. His research involves Multicast Routing, Reliable Multicast, High Speed Computer Networks and e-Governance. He has published more than 20 research papers in various International Journals and Conferences. He has also authored a book entitled “Design and Analysis of Algorithm” and a book chapter entitled “Transport Protocols in Wireless Sensor Networks”. 

Surendra Sai University of Technology, Burla, India. His research involves Multicast Routing, Reliable Multicast, High Speed Computer Networks and e-Governance.