CHAPTER 3

NODE IDENTITY ASSESSMENT

A malicious node may impersonate as a genuine node by using its node identity. Such nodes are called replica nodes. Replica node detection is performed to identify these nodes and revoke the nodes. Replica node detection may be probabilistic or deterministic methods. Most of the methods presented in literature survey have probabilistic nature of detection. In probabilistic methods replica node detection may not be guaranteed. In deterministic methods replica node detection may be accurately determined. In this chapter Replica Node Detection Based on Localization technique (RDBLT) and RFID (RDBRFID) is presented. Both the methods are performed in cluster environment. The cluster head is responsible for replica node detection. The replica node detection is performed in three levels: Intra cluster, inter cluster and overlapping cluster. Replica nodes are intra when they are inside a cluster. When replica nodes are in two or more clusters, they are called inter cluster replica nodes. Replicas in overlapping clusters are overlapping replica nodes. Figure 3.1 presents the replica nodes in overlapping cluster environment. Cluster heads A, B and C share their cluster member details to detect replica nodes. R1 replica nodes is an example for intra cluster replica nodes in Cluster A. R1 replica nodes are present in both cluster A and C and can be detected in inter cluster replica node detection. In overlapping clusters replica nodes shared between clusters are called boundary nodes. R1 node is a boundary node being shared between Cluster A
and B. It may need verification to confirm that it is a genuine node and is being shared between clusters.

Figure 3.1 Replica nodes in a cluster

In wired networks, a node can be uniquely identified through its MAC address. However, in WSN, the MAC address of the nodes can be duplicated or spoofed easily (Mathieu Cunche et al. 2013). The process of identifying the node uniquely in a WSN is a challenging task. The existing addressing scheme proposes a method of assigning address to the neighboring nodes and that in turn would assign address to its own neighboring nodes (Jobin et al. 2004). However, there is no mechanism to ensure the uniqueness of the assigned address. Hence, this does not provide solution for the problem of replication identification. The proposed method RDBRFID uses a novel approach of replica identification for overlapped clustered in WSN, based on unique addressing using RFID (Jain et al. 2010). This system uses RFID based addressing scheme where a node needs to incorporate RFID tag into it for its unique identification. The RFID address is unique and cannot be
Another proposed method RDBLT uses Localization technique based on RSSI and Triangulation algorithm to compute the locality of nodes in the clusters for replica detection. The architecture of the proposed system is presented in Figure 3.2.

![Figure 3.2 Architecture of node identity assessment](image)

### 3.1 LOCALIZATION

The proposed method RDBLT uses localization method to find the location of the nodes in the clusters and the results obtained are used for replica detection. The system uses triangulation method combined with the Received Signal Strength Indicator (RSSI) to find the distance between a known node and the unknown node (Mao G et al. 2007). The position of the unknown node can be determined using the obtained distance. Three anchor nodes are necessary to find the location of the unknown node. The Euclidean distance from the three anchor nodes to the unknown node in triangulation method, the intersection of the three anchor nodes gives the most accurate location of the node. Figure 3.3 depicts the unknown node and the anchor nodes location requirements. The unknown node is in the coverage area of the three anchor nodes. The distances from the unknown node to the anchor
nodes are computed using the RSSI. Using the computed distances and the anchor node locations, the location of the unknown node is determined. The algorithm for location computation using triangulation and RSSI is presented in Figure 3.4.

Figure 3.3 Location requirements of unknown and anchor nodes

3.1.1 A Scenario for Location Computation

The coordinates (x,y) of unknown node N are to be found. The coordinates of Anchor nodes A1,A2 and A3 are available as (x1,y1),(x2,y2) and (x3,y3). The RSSI values of each anchor node is made available. Using RSSI values the distance (d1,d2 and d3) of unknown node and anchor nodes (A1,A2 and A3) is computed. Using the computed distances (d1,d2 and d3) and the coordinates of the anchor nodes the unknown coordinate is computed.

According to Friis transmission equation in free space radio propagation model (Mao G et al. 2007),

Let RSSI from A1=9.865e^{-09}, P_t = 0.2828, G_t =1.0 , G_r =1.0 , \lambda=1 , L= 1

\[ d^2 = P_t \times G_t \times G_r \times \lambda^2 / (4\pi)^2 \times L \times P_r \]  

(3.1)
where \( d \) is the distance, \( P_t \) is the antenna transmission power, \( P_r \) is the antenna receiving power, \( G_t \) is the transmission gain and \( G_r \) is the receiver gain, \( \lambda \) is the wavelength and \( L \) is the loss.

Substituting values in Equation 3.1 (Jobin et al. 2004) Equation 3.2 is obtained as

\[
d^2 = (0.2828)(1.0)(1.0)(1)^2/(4(3.14)(1)(9.865e^{-09}))
\]  

(3.2)

d is obtained as 1.21185 by solving Equation 3.2. The RSSI value from three anchor nodes to a unknown node and their corresponding distances are given in Table 3.1.

| Input: coordinates \([(x_1, y_1), (x_2, y_2), (x_3, y_3)]\) of anchor nodes \([a_1, a_2, a_3]\), nodes \([n_1, n_2, ..., n_n]\) with unknown coordinates |
| Output: coordinates \([n_1, n_2, ..., n_n]\) |
| Step 1: Start |
| Step 2: Known nodes \(KN = [a_1, a_2, a_3]\) |
| Step 3: Get RSSI values from \([n_1, n_2, ..., n_n]\) where \(n_1, n_2, ..., n_n\) are neighbor nodes of \(KN\). |
| Step 4: The node \(u\) is selected from \(n_1, n_2, ..., n_n\) which has the strongest RSSI from three nodes in \(KN\). |
| Step 5: Distance \(d_1, d_2, d_3\) is computed using RSSI |
| Step 6: Using \(x_1, y_1, x_2, y_2, x_3, y_3\) and \(d_1, d_2, d_3\), compute \((x_u, y_u)\) of node \(u\) is found |
| Step 7: Add \(u\) to \(KN\). |
| Step 8: Repeat the steps from 4 to 8 until all coordinates of nodes in \([n_1, n_2, ..., n_n]\) are found |
| Step 9: Stop |

**Figure 3.4** Algorithm for location computation using triangulation and RSSI
Table 3.1 RSSI, co-ordinates (x, y) and distance from the anchor nodes to the unknown node

<table>
<thead>
<tr>
<th>Anchor nodes</th>
<th>RSSI</th>
<th>Distance</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>A₁</td>
<td>9.865e⁻⁹</td>
<td>1.21185</td>
<td>109.8</td>
<td>504.8</td>
</tr>
<tr>
<td>A₂</td>
<td>3.668e⁻¹⁰</td>
<td>3.2752</td>
<td>301.9</td>
<td>483.3</td>
</tr>
<tr>
<td>A₃</td>
<td>6.784e⁻¹⁰</td>
<td>5.8031</td>
<td>265.3</td>
<td>323.6</td>
</tr>
</tbody>
</table>

According to the Euclidian distance formula, the distance between two points \((x₁, y₁)\) and \((x₂, y₂)\) (Mao G et al. 2007) in a plane is given by

\[
d₁((X, Y), (x₁, y₁)) = \sqrt{(X - x₁)^² + (Y - y₁)^²}
\]  (3.3)

\[
d₂((X, Y), (x₂, y₂)) = \sqrt{(X - x₂)^² + (Y - y₂)^²}
\]  (3.4)

\[
d₃((X, Y), (x₃, y₃)) = \sqrt{(X - x₃)^² + (Y - y₃)^²}
\]  (3.5)

Resolving the equations (3.3), (3.4) and (3.5) in matrix form, Equation (3.6) is obtained.

\[
2 \begin{bmatrix} (x₃ - x₁)(y₃ - y₁) \\ (x₃ - x₂)(y₃ - y₂) \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} (d₁² - d₃²) - (x₁² - x₃²) - (y₁² - y₃²) \\ (d₂² - d₃²) - (x₂² - x₃²) - (y₂² - y₃²) \end{bmatrix}
\]  (3.6)

Equation 3.6 is solved to obtain \(X, Y\) by substituting the values of \((x₁, y₁), (x₂, y₂), (x₃, y₃), d₁, d₂\) and \(d₃\) values available in Table 3.1. After substitutions in Equation 3.6, Equations 3.7 and 3.8 are obtained.
Solving the linear Equations 3.7 and 3.8, \((X, Y)\) is obtained as \(X=198.2, Y=423.0\). The location \((X, Y)\) of the unknown node \(N\) is \((198.2, 423.0)\). The computed location \((X, Y)\) is used along with anchor nodes locations \((x_a, y_a)\) to compute the locations of the rest of the unknown nodes.

### 3.2 BLOOM FILTER

Bloom filter is a space efficient probabilistic data structure based on hashing techniques used for forming the cluster member list (Geravand et al. 2013). Bloom Filter is used in the proposed system because of its space efficiency and easier querying techniques for memberships. The cluster head of every cluster is responsible for the preparation of the list. Figure 3.5 depicts the Bloom filter list, prepared by the cluster heads which is forwarded to other cluster heads. This would help the cluster heads to find the nodes that are common to more than one cluster.
Consider a set of Cluster Members $CM = [cm_1, cm_2, cm_3...cm_n]$ which belongs to a cluster head CH. The CH calculates the hash value $H$ for each CM. The bloom filter BF array has $n$ bits as output. To insert each $CM_i$ into BF, the hash value of CMs, $H [h_1, h_2, h_3,h_n]$ for $CM_i$ is obtained. The corresponding bits $BF (h_i)$ are set as 1. All the other bits in BF remain 0. To query whether a CM is present in BF, compute the $H$ value of CM, then check the corresponding position bits for 1 in BF. If all the bits are 1, then it is proven that the CM is present in BF. Figure 3.6 illustrates the steps of insertion and Figure 3.7 illustrates the steps required to query in Bloom Filter.
Input: A set of n Cluster members $CM = \{ cm_1, cm_2, cm_3, ..., cm_n \}$

Output: Bloom Filter Array BF.

Step 1: BF={0,0,......}

Step 2: for i=0 to n

  { a=cm_i

  for j=1 to k //hashing of cmi is performed k times

    { b=h_j(a);
      a=b
    }

    $H_i=a$

    BF($H_i$) = 1 }

Figure 3.6 Algorithm for insertion of node id

Input: A node-id cm and Bloom Filter Array BF;

Output: Boolean True or False.

Step 1: Initialize a=cm

Step 2: for j=1 to k

  //hashing of cmi is performed k times

  { b=h_j(a)
    a=b
  }

  H=a

Step 3: If BF(H) is 1

  Return True

Else

  Return False

Figure 3.7 Algorithm to query for node id

According to the working principle of the Insertion algorithm, the cluster members $CM_i$ of a cluster will not be left unmarked in a bloom filter. Hence there can be no chance for false negatives in bloom filter. However, there are possibilities for some of the non-cluster member nodes to be marked as cluster members in the Bloom filter. Hence false positives are possible. In
order to restrict the false positive probability, certain considerations in parameters have to be carried out.

The false positive probability in a bloom filter (Geravand et al. 2013) is given in Equation 3.6.

\[ p = (1 - (1 - 1/m)^{kn})^k \]  \hspace{1cm} (3.6)

Where \( m \) is the number of bits in the Bloom filter array, \( k \) is the number of times the hash functions is executed to generate the hash value of the CM id and \( n \) is the total number of nodes in a cluster.

Equation (3.5) can be further reduced as

\[ p = (1 - e^{-kn/m})^k \]  \hspace{1cm} (3.7)

The number of hash functions \( k \) can be given as

\[ k = \frac{m}{n} \ln 2 \]  \hspace{1cm} (3.8)

Substituting \( k \) in (3.6),

\[ p = (1 - e^{-\left(\frac{m}{n} \ln 2\right)} \frac{m}{n} \ln 2)^m \ln 2 \]  \hspace{1cm} (3.9)

Equation (3.8) is further simplified as

\[ \ln p = -\frac{m}{n} (\ln 2)^2 \]  \hspace{1cm} (3.10)

From Equation (3.9), \( m \) can be obtained as

\[ m = \left\lfloor \frac{n \ln p}{(\ln 2)^2} \right\rfloor \]  \hspace{1cm} (3.11)

The false positive probability \( p \) is very small for well-chosen parameters (\( k, n, m \)) where the parameters \( n \) and \( m \) are to be larger and \( k \) to be small. Given a network of \( n = 1500 \) nodes, with an average degree of a node \( d=10 \), hop count \( h=2 \), number of clusters or cluster head \( c = n/d^h = 15 \), average number of cluster members is 100. Size of the bloom filter, \( m \) is chosen as 1200 bits and \( k \) is chosen as 7, so that the false positive probability, \( p \) is limited to 2%.
### 3.3 ADVERSARY MODEL AND ASSUMPTIONS

The purpose of the adversary system is to replicate or clone the existing nodes in the network. These replicated or cloned nodes can behave like genuine nodes and may try to attack other nodes. The proposed system considers multi-hop homogenous wireless sensor network where all nodes are considered to be alike. The nodes are assumed to have unique ID. Except some anchor nodes, all the nodes are location-unaware. The system does not require base station or a system to coordinate the activities of subsets of nodes because of its decentralized and distributed nature. The nodes organize themselves to arrive at collective decisions and require cooperation among them. The system assumes that the sensor nodes are stationary and they transmit at the same power level and transmission range. The proposed system also assumes that there is underlying security mechanism in the proposed network model. Existing cryptographic algorithms are safe to encrypt and decrypt data (Dnayang Qin et al. 2016).

Similarly existing methods for key exchange and building may be used. The proposed system also assumes that there are existing routing techniques (Zhenquan Qin et al. 2013). The proposed method also requires the number of genuine nodes to be more than 50% of the total number of nodes in the network. These assumptions are made with the expectation that these choices do not affect the results of the proposed system.
3.4 REPLICA DETECTION SCENARIO

Each cluster head creates a list of its nodes and forms the Bloom Filter. The Bloom filter is propagated to all the cluster heads in the network. Each cluster head performs the analysis of the results of bloom filter and detects for replicated nodes in its cluster or common nodes between clusters. A transaction revoke message is forwarded to all the clusters on identification of replica nodes. The cluster head would verify the genuineness of the received Bloom filter based on the following scenarios.

**Case 1: Cluster Member may be replica node**

Replica nodes existing within a cluster is said to be an intra cluster replica whereas replicas in two or more clusters are inter cluster replicas. A Cluster head CHx is responsible for Intra cluster replica detection. When the cluster head collects the cluster member details to form the Bloom Filter, verification is done by using RFID in RDBRFID or by using location computations in RDBLT. Hence Intra cluster replicas are ruled out in the initial stage. The replicas detected will be alerted to other cluster heads also. In the case of inter cluster replica, it can be detected with the Bloom Filter list received from the corresponding Cluster head. In this case, the cluster head (CHx) generates the current Bloom filter list with the current cluster members. This list is compared with the received list from another cluster head (CHy). If both the lists contain common nodes, there are two possibilities: the node may be a shared node or a replica node. Further, the replica node is detected by applying the algorithms for replica detection using RDBRFID or RDBLT. These two methods are implemented for the verification of the nodes. The steps of RDBRFID technique are explained with the flow chart presented in the Figure 3.8.
Figure 3.8 Steps for replica node detection
Case 2: Cluster Head may be replica node

In the case of cluster head having acted as a replica node, it would not have given the correct list of cluster members using bloom filter. In this case, the cluster head (CHx) selects a random cluster member (CM) from the cluster, whose bloom filter has to be verified. The CHx would make the CM to regenerate its new bloom filter list and compares it with the received list from another cluster head (CHy). If there is difference in comparison, the process of verification is repeated for ‘n’ number of times where n is a predefined value. The value of ‘n’ is based on the security requirements of the application. The cluster head CHx would alert about the replica of the cluster head CHy to other cluster heads if the verification process fails. These steps are illustrated using the flowchart presented in Figure 3.9.
3.5 COMMUNICATION COMPLEXITY

The bloom filter message exchange occurs between the cluster heads. Hence the communication complexity of the system mainly depends on the number of cluster heads. Assuming the number of nodes in the network as \( n \), the number of cluster heads as \( c \), the number of cluster members as \( cm \), cluster head exchanges as \( 2(c-1) \) messages of \( b \) bit messages, the communication complexity can be summed as \( O(c^2) \).

In N2NB, a node broadcasts its neighbor id and its location claim to the neighboring nodes. The neighboring nodes broadcast the received details to their neighbors. A node detects a replica when it finds conflicting claims. As in N2NB, DM forwards its location claims by broadcasting. But the replica detection is performed by a set of nodes, called witness nodes. Witness nodes perform the monitoring process. Since both the methods use broadcasting approach for replica detection, the communication complexity is \( O(n^2) \). In LSM, a node forwards a received location claim to a selected node. The node selection is based on rumor routing. As the location claim gets forwarded in a particular path, conflicting claims can be detected at intersection points of the path. The communication complexity in LSM is \( O(n\sqrt{n}) \). In N2NB, DM and LSM, the received node ids and location claims need to be sent by the nodes. In proposed methods, RDBRFID and RDBLT, the Bloom Filter is used. The data containing entire node id of a cluster is sent in the form of bits as a single list between cluster heads. The communication complexity of the proposed approaches is less when compared with N2NB, DM and LSM. The communication complexity of the proposed and the existing methods is represented as is Equation 3.12.

\[
n^2(|id|) > n\sqrt{n(|id|)} > c^2(m) \quad (3.12)
\]
where \( n^2 |\text{id}| \) is the communication complexity of N2NB and DM, \( n \) is the number of nodes and \( |\text{id}| \) is the id and location claims.

\[
n^\sqrt{n(|\text{id}|)} \text{ is the communication complexity of LSM, } n \text{ is the number of nodes and } |\text{id}| \text{ is the id and location claims.}
\]

\( c^2 (m) \) is the communication complexity of the proposed approaches RDBRFID and RDBLT, where \( c \) is the number of cluster heads and \( m \) is the number of bits of Bloom Filter.

\[
m \text{ can be determined using Equation 3.13 as}
\]

\[
m = - \frac{n \ln p}{(\ln 2)^2}
\]

(3.13)

It is noted that \( c \) is less than \( n \), hence it may be concluded that when

\[
c \leq \frac{n|\text{id}|(\ln 2)^2}{-\ln p}, \text{ the proposed systems are efficient when compared to LSM.}
\]

When \( c \leq \frac{n^2|\text{id}|(\ln 2)^2}{-\ln p} \), the proposed methods are efficient than N2NB and DM. Using these parameters, communication cost in the proposed method for 1500 nodes with 15 clusters and 100 cluster members is 15 bloom filters of 1200 bits in length. Since Bloom filter has compressed the data about entire cluster members into a single array, the communication cost is minimized. Further communication may be required for replica confirmation which is negligible.

### 3.6 STORAGE COMPLEXITY

Each CH sends its BF which is \( b \) bits in length to other peers or cluster heads in the network. With respect to a CH, it needs to store its own BF and the other peer CH’s BF. Hence \( c \) number of BF will be stored in a CH. Thus the storage cost can be summed to be of the \( O(c) \). A major aspect of Computation cost in the cluster heads is that the proposed system requires
hash computations. The performance of SHA-1 on a Pentium 4 is equal to 11.4 cycles per byte. The computation of SHA-1 is slow on AVR, 8 bit micro controllers and the computation requires 1183 CPU cycles. Compared to the time required for Public key cryptography, the amount of CPU time for SHA-1 is less (Nithin Chandran et al. 2016). Other light weight hash functions can be considered as optional to SHA-1 (Knuth et al. 1997).

Table 3.2 presents the comparison of communication cost, the storage cost and the number of nodes involved in replica detection in various existing and proposed methods. In RM, DM, LSM, N2NB, SDC, P-MPC, RED and RDE, the number of witness nodes are the nodes involved in replica detection. In the proposed method, the storage cost and the communication cost are based on the number of cluster heads.

<table>
<thead>
<tr>
<th>No.</th>
<th>Method</th>
<th>Communication Cost</th>
<th>Storage Cost</th>
<th>R</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>RM, DM</td>
<td>$O(n^2)$</td>
<td>$O(\sqrt{n})$</td>
<td>$\sqrt{n}$</td>
<td>$n$- number of nodes</td>
</tr>
<tr>
<td>2.</td>
<td>LSM</td>
<td>$O(n\sqrt{n})$</td>
<td>$O(\sqrt{n})$</td>
<td>$\sqrt{n}$</td>
<td>$n$- number of nodes</td>
</tr>
<tr>
<td>3.</td>
<td>N2NB</td>
<td>$O(n^2)$</td>
<td>$O(d)$</td>
<td>$D$</td>
<td>$d$- number of neighbor nodes</td>
</tr>
<tr>
<td>4.</td>
<td>RDE</td>
<td>$O(n\sqrt{n})$</td>
<td>$O(d)$</td>
<td>$D$</td>
<td>$n$- number of nodes, $d$- number of few selected neighbor nodes</td>
</tr>
<tr>
<td>5.</td>
<td>SDC</td>
<td>$O(s.n)$</td>
<td>$O(s.ps)$</td>
<td>$S$</td>
<td>$s$- no. of cells, $n$- no. of nodes, $ps$- probability that a node hears a location claim</td>
</tr>
<tr>
<td>6.</td>
<td>P-MPC</td>
<td>$O(s.n)$</td>
<td>$O(s.ps)$</td>
<td>$S$</td>
<td>$s$- no. of cells, $n$- no. of nodes, $ps$- probability that a node hears a location claim</td>
</tr>
<tr>
<td>7.</td>
<td>RED</td>
<td>$O(g.p.d.\sqrt{n})$</td>
<td>$O(g.d.p)$</td>
<td>$(g.d)$</td>
<td>$g$- no. of witness nodes, $p$- probability of hearing location claims, $d$- distance between nodes, $n$- number of nodes</td>
</tr>
<tr>
<td>8.</td>
<td>RDBRFID, RDBLT</td>
<td>$O(c^2)$</td>
<td>$O(c)$</td>
<td>$C$</td>
<td>$c$- number of clusters or cluster heads</td>
</tr>
</tbody>
</table>
3.7 SIMULATION RESULTS

A set of simulations is made to run in NS2 to compare the proposed system with the protocols LSM and RM (Parno B et al. 2005). The tests are performed to find the detection rates, communication overheads and the energy gain. The tests are performed with the assumption that the cryptographic layer is not taken into consideration. Also the cost of cluster construction is not considered. The system is compared to a non-cluster environment as in Fault Tolerant Virtual Back Bone Trees (FTVBT) (Suganthi et al. 2015) and k-coverage networks (Xingwei Wang et al. 2010). FTVBT is a method which focuses on fault tolerant energy management in WSN. K coverage focuses on providing coverage to all nodes connected in a network. In both the methods, RFID and Localization techniques are implemented as FTVBT- RDBRFID, FTVBT-RDBLT, K-coverage RDBRFID, K-coverage RDBLT to study the performance of the methods in non-clustered environments. The parent node performs the replica detection in FTVBT and the normal nodes perform the replica detection in distributed K-coverage networks. The number of replica nodes is varied between 1 and 30.

3.7.1 Simulation Parameter

The Bloom filter size is computed dynamically by each CH according to the number of CM. It is also influenced by the false positive probability to be less than or equal to 2%. The optimal value for k is assumed to be 7. It is the number of times the hash function is called by the algorithms for bloom filter construction. Both the replica node detection methods RDBRFID and RDBLT are performed with a hop count h =2. For RM, p is assumed as 0.15 and g is set such that pxdxg = √n and six lines are used for LSM. The tests are performed using IEEE 802.11 physical and Mac Layers in NS2. Each simulation is run with n nodes, where n ∈ [100; 500] distributed randomly over a square field of 500 x 500 m² with degree d for a node,
where $d \in [15; 45]$. The average probability of detection for a single node replication is done as defined in (Parno B et al. 2005). It represents the number of times; the protocol must run to detect the replication attack. The comparison of communication cost and energy gain consumption is presented for performance analysis. It should also be noted that in Figure 3.10, Figure 3.11 and Figure 3.12, hop count $h$ is 2. In FTVBT system, the non-tree nodes join the parent node to form tree structure based on transmission range. The transmission distance $d$ is set as 30m. Polyhedral structure is formed in K-coverage networks for better communication capabilities (Xingwei Wang et al. 2010).

3.7.2 Results of One Replica Node

The average detection probability of a single node replication is presented in Figure 3.10. This probability is 100% in the case of RDBRFID and is 98 in RDBLT. It is 75% for the RM and LSM protocols. The obtained probability is in accordance with the one described in (Parno B et al. 2005). The probability is equal to 1 in the case of RDBRFID because of its deterministic nature in identifying replicas, rather than being probabilistic. Any replica will be detected by any cluster head in the process of bloom filter verification. But both proposed methods may fail when the entire cluster is replicated. This is due to the failure in bloom filter verification, as it is purely dependent on the cluster head and a random cluster member. In case of cluster replication, all cluster members may agree to the bloom filter sent by the cluster head. But it is assumed that such a case is highly unlikely to occur. Both the proposed systems are efficient in terms of average probability of detection of replicas when compared with the existing methods in (Parno et al. 2005). The proposed systems implemented in FTVBT and K-coverage networks show similar detection probabilities when compared with RDBRFID and RDBLT methods.
Figure 3.11 presents the average number of packets sent and received per node for the proposed and existing algorithms. It is evident from the data that RM algorithm generates more traffic and is less efficient when compared with the other methods. The methods k-coverage-RDBRFID and k-coverage-RDBLT also generate more traffic due to their underlying protocol working principle. In k-coverage networks, node’s degree is maintained to be a minimum of k nodes, so that the nodes are within the coverage of at least k nodes. A node will be monitored by k nodes and this process requires communication between the nodes and hence accounts for more communication overhead. The methods RDBRFID and RDBLT generate lesser traffic than these protocols because these methods require lesser communication. This is because the detection and verification process takes place in cluster heads; hence communication cost is also limited to the number of cluster heads. In addition some negligible communication takes place with randomly selected nodes for bloom filter verification. It is also observed that the systems based on FTVBT also need similar less communication. It is due to the fact that the communication pattern in all
these methods is similar. In FTVBT, parent node communicates its bloom filter (its id || child node ids) to its parent. The same method continues among all parent nodes till the sink node is reached.

**Figure 3.11 Communication overhead for proposed and existing methods**

In WSN, minimizing the energy consumption of each node is an important criterion. Figure 3.12 presents the comparison of Energy consumption gain between the existing and the proposed methods. Energy gain is computed using the energy ratio between existing and proposed methods. $E_{\text{LSM}}/E_{\text{RDBRFID}}, E_{\text{RM}}/E_{\text{RDBRFID}}$ and $E_{\text{Localisation}}/E_{\text{RDBRFID}}$ are computed and the comparative study is performed. $E_{\text{LSM}}$ is the energy consumption in LSM algorithm and $E_{\text{RDBRFID}}$ is the energy consumption in the proposed RFID algorithm. This metric is considered as it accounts for each bit sent and received by each node. It also takes into account the energy spent for computations at each node. The energy gain varies from 1.3 to 3.4, according to the number of nodes considered. This leads to the proposition that the proposed method RDBRFID is at least 1.3% and 1.6% more energy efficient.
than the existing LSM and RM algorithms. Similarly when both the proposed methods are compared, it is found that RDBRFID method is 1.3% more energy efficient than RDBLT. It is because the RDBLT method requires more energy to be spent in computing the locality of the nodes, to find whether it is genuine or replica node. Hence the RDBRFID algorithm is found to have better energy gain in comparison with LSM, RM and RDBLT methods.

![Bar graph showing energy gain of RDBRFID compared with LSM, RM, and RDBLT](image)

**Figure 3.12  Energy gain of RDBRFID compared with the LSM, RM and RDBLT**

### 3.7.3 Results of Varied Number of Replica Nodes

The detection probability of proposed methods RDBRFID and RDBLT for 2 number of replica nodes is presented in Figure 3.13. The replica nodes may be normal sensor nodes or the cluster heads. The hop count h is varied from 1 to 2 and results are obtained. The other simulation parameters are the same as in section 3.12. All the replicated nodes are placed randomly in the network at the beginning of the simulations. For varying h, the proposed methods offer detection probability of above 96%.
Figure 3.13 Detection probabilities for proposed methods: 2 replica nodes

The average detection probability of the proposed methods RDBRFID and RDBLT for 15 replica nodes is presented in Figure 3.14. The methods RDBRFID and RDBLT for varying h, offer better detection probability of above 96%. It is also noted that the detection probability rate of the proposed methods, decreases with an increase in the number of replica nodes. This is due to the bloom filter verification step failure, if replica nodes are selected in the process of verification. It must be noted that when hop count h increases, the size of a cluster increases and thereby number of cluster heads decrease. A decrease in the number of cluster head decreases the communication overhead.

Figure 3.14 Detection probabilities for proposed methods: 15 replica nodes
Increasing the number of replica nodes, the average replica detection rate is found and the detection rates of both the proposed methods RDBRFID and RDBLT are presented in Figure 3.15. The number of nodes is 100, \( h \) is 1 and 2 and the replica nodes are varied from 5 to 30. The average probability of detection rate is found to be 98\% for RDBRFID and 97\% for RDBLT.

Figure 3.15 Average Probability of detection rate for proposed methods

Figure 3.16 presents the probability gain for comparing the two proposed methods RDBRFID and RDBLT. The results are obtained for varying \( h \) values 1 and 2. It is observed that the probability gain of RDBRFID over RDBLT varies from 1.01\% to 1.02\%. It is comprehended that this difference is because of the deterministic nature of the proposed method RDBRFID when compared with the RDBLT method.
Replica node detection in overlapping clusters of WSN is difficult as node id may be shared between clusters. The proposed system uses two methods to identify replica nodes. The first system uses node’s location for replica node detection. The second method uses unique node identity (RFID) for the same. The methods are compared with the existing systems RM and LSM and their performance is analyzed. The methods are further observed in non-cluster environments FTVBT and K-coverage WSN for performance analysis and the results are found. It is observed that the proposed methods RDBRFID and RDBLT offer minimal communication overhead when compared with the existing methods.

Figure 3.16 Probability gain of RDBRFID over RDBLT

3.8 SUMMARY