CHAPTER 4

POWER EFFICIENT CONTEXT AWARE
BROADCASTING PROTOCOL

4.1 INTRODUCTION

In MANET, broadcasting is a common operation for route establishment and sending control and emergency messages. The emergency messages have to be broadcast with a high priority whereas a few control messages may be broadcast with a low priority. The main objective of any broadcasting algorithm is to reduce the number of rebroadcasts, keeping the bandwidth and computational overhead as low as possible. Context information can be used effectively to prioritize the messages before broadcasting. The situation of an entity can be portrayed using the context information like communication between a user and an application. The detections about the current situation of the user can be processed by the computing system that creates awareness of the user’s context.

The hardware sensors such as GPS receiver (for positioning), bluetooth equipment, software sources like the agenda of a person and the list of the currently opened documents can be used to detect the context. Based on the sensor data, the context information is changed in parallel for active context awareness. The application of the active context awareness is the automatic change in date and time when the user switch on the mobile phone. In the passive context aware applications, the context information is presented to the user and the changes in the updation are specified by the user.
An example of the passive context aware application is that before changing the date and time, the user may be asked to decide whether the date and time has to be updated or not.

Any Context Aware Broadcasting Protocols (CABP) work to achieve the following design objectives.

- The protocols should work well on MANET, given all the challenges they involve.
- The protocols should be able to provide the user with the ability to control its behavior for each message.
- The information about the network topology is mandatory to design the protocol in an efficient manner.

The protocol calculates the delivery probability for each node based on the message urgency. The broadcast message is classified as urgent based on the real time application by the underlying MAC protocol. The attribute urgency is modeled as a function in the domain $[0, 1]$. It considers two extreme values, 0 for lower bound and 1 for upper bound to calculate the message urgency.

When the message urgency is set as 0 in the lower bound then the node speed and delivery ratio requirements are not considered. The resource utilization must be less when the message urgency is minimum. The delivery probability of the forwarding messages is decreased when there is an increase in the number of the neighbors that help in minimizing the resource utilization. The message urgency is set as 1 for the upper bound and the messages have to be broadcast without any limitations. The broadcast must be fast and the delivery ratio must be 100%.
Despite the utilization by other applications, the available resources are utilized effectively. Using the probability of 1, the broadcast messages are forwarded by the nodes to all their neighboring nodes. An associative table that has node ID (which transmits the message) and the neighbor’s node ID is created. The table is referred before forwarding the message to any new node. The random access delay is used by the broadcasting protocol to allow the nodes to avail “wait before send” mechanism. When a broadcast message is received by a node, it waits for a while to check the associative table and then forwards the information to the next node. If a neighbor’s ID is yet to be stored in the set, the messages are allowed to be forwarded.

The major drawbacks in the Context Aware Broadcasting Protocol are

- The delivery probability is calculated based only on message urgency to forward the message.
- It does not consider other parameters such as location, trust level, Quality of Service (QoS) requirements and message reliability for context awareness.
- There is no guarantee that the broadcasting process will reach every node.
- The protocol is not energy efficient since the power control mechanisms are not considered.

4.2 SYSTEM MODEL

A new Power Efficient Context Aware Broadcasting Protocol (PECABP) is proposed based on the Delivery Probability (DP) to overcome the drawbacks in CABP protocol with a view to maximize the number of nodes reached, minimize the duration of the broadcasting process and the bandwidth utilization.
4.2.1 Overview

Initially, the delivery probability of each node is determined based on its connectivity, power level and trust index and each node maintains a table of its neighbor’s delivery probability. The message is broadcast based on its priority and the header field of the broadcast message is given in Figure 4.1.

<table>
<thead>
<tr>
<th>ID</th>
<th>ID</th>
<th>ID</th>
<th>...</th>
<th>ID</th>
<th>ID</th>
<th>CF</th>
<th>0.8</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID of source node</td>
<td>ID’s of neighboring nodes</td>
<td>Flag</td>
<td>Urgency</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.1 Message Header with Flag Values

The message header has a context flag (CF) with three values namely, the urgency U, the reliability R and the normal N. Whenever a new message arrives with the flag value set as U, it has to be delivered in a short time but it can tolerate the packet losses up to a small extent. The message having the flag value R has to be delivered without any loss but with a tolerable delay. The other messages having the flag value N are treated as normal that can tolerate both the delay and packet loss.

When the flag value is U, the message is forwarded to the neighboring nodes with the condition DP > Th₁ and when the flag value is R, the message is forwarded to the node that satisfies the condition DP > Th₂ where Th₁ and Th₂ are the threshold values of DP. Also to ensure a reliable message delivery and to avoid packet dropping, redundancy is added to the packets of the message using the forward error correction technique for the flag value R. When the flag value is N, the message is forwarded to the node with a minimum DP.
4.3 PECABP PROTOCOL

The protocol works by calculating the delivery probability of each node from node’s connectivity, trust index and power level. The connectivity of the node is determined based on its relative mobility and the trust index of the node can be estimated from the successful forwarding or delivery history of that node. The Power level is obtained directly from the battery of the node.

4.3.1 Estimation of Connectivity

The power level detected at the receiving node $Pr$ is the indication of the distance between the transmitting and receiving node pairs. Friis’ free space propagation model is used to calculate the node’s connectivity. The model is used in telecommunications engineering and gives the power received by one antenna under idealized conditions given another antenna some distance away transmitting a known amount of power. It specifies that the ratio of the receiver power and transmitter power is inversely proportional to the square of the physical distance between the transmitter and the receiver, i.e,

$$ \frac{Pr}{Tr} = \frac{1}{d^2} \quad (4.1) $$

The relative mobility between the two nodes can be calculated from the ratio of $Pr$ between two successive packet transmissions from a neighboring node by periodically sending “Hello” messages. Hence, the relative mobility metric $Mr_{L}(K)$ is determined for any node ‘L’ with respect to the neighboring node ‘K’ using the following equation.

$$ Mr_{L}(K) = 10 \log_{10} \frac{Pr_{L \rightarrow K}^{new}}{Pr_{L \rightarrow K}^{old}} \quad (4.2) $$
When the value of $P_{L ightarrow K}^{\text{new}} < P_{L ightarrow K}^{\text{old}}$, $M_{R}(K)$ is negative. The negative value of the relative mobility metric between any two nodes indicates that the two nodes are moving away with respect to each other. On the other hand if $P_{L ightarrow K}^{\text{new}} > P_{L ightarrow K}^{\text{old}}$, then $M_{R}(K)$ is positive and indicates that the nodes are moving closer to each other.

For a node with ‘m’ neighbors, ‘m’ numbers of $M_{R}(K)$ values are available. The aggregate relative mobility value $M_{R}(L)$ of any node ‘L’ is determined by calculating the variance (with respect to zero) of the entire set of relative mobility values $M_{R}(K_{i})$, where $K_{i}$ is the list of neighbors of L. It is given in Equation 4.3.

$$M_{R}(L) = \text{var}_{0}\left(M_{R}(K_{1}), M_{R}(K_{2}), ... M_{R}(K_{m})\right)$$

$$= E\left[(M_{R}(L))^{2}\right]$$

(4.3)

Var$_{0}$ denotes the variance with respect to zero (and not the mean of the sample) and is equal to $E\left[(M_{R}(L))^{2}\right]$ which is the expected value of the squares of the m relative mobility samples from all the neighbors of a node ‘L’. In this manner, any node L which also acts as a receiver measures the power levels in the successive transmissions from all of its neighbors and a variance of these values (with respect to zero) is a representative value for the aggregate relative mobility metric $M_{R}(L)$ for that node.

When the aggregate relative mobility metric $M_{R}(L)$ of a node ‘L’ is low, the node ‘L’ is relatively less mobile with respect to its neighbors. On the other hand, when the aggregate relative mobility metric $M_{R}(L)$ of a node ‘L’ is high, the node ‘L’ is highly mobile with respect to its neighbors. Using this
aggregate relative mobility metric, the connectivity $C_N$ of a node is estimated using Equation 4.4. As the mobility decreases, the connectivity keeps on increasing.

$$C_N = \frac{1}{M_{rL}(K_1)}$$

(4.4)

### 4.3.2 Estimation of Trust Index

Let the topology $S \rightarrow N1 \rightarrow N2 \rightarrow N3 \rightarrow D$. Let $\{T_{i1}, T_{i2}, \ldots\}$ be the initial trust indices of the nodes $\{n1,n2,\ldots\}$ respectively along the route $R1$ from the source $S$ to the destination $D$. An additional data structure called Neighbor’s Trust Index Table (NTIT) is maintained in each node of the network. Initially, when a source node $S$ wants to establish a route to the destination $D$, it sends the Route Request (RREQ) packet. The creation of the neighbor’s trust index table is given in Figure 4.2.

![Figure 4.2 Creation of Neighbor’s Trust Index Table (NTIT)]
Each node keeps track of the number of packets it has forwarded in the given route using a Forward Counter (FC). When a node $N_2$ receives a packet from a node $N_1$ in the same route, the node $N_2$ increments the forward counter of node $N_1$ by $FC_{N_1} = FC_{N_1} + 1$. The Neighbor’s Trust Index Table (NTIT) of node $N_2$ is modified using the values of $FC_{N_1}$. The NTIT table is attached along with the RREQ packet. Similarly, each node updates its NTIT table using the FC values and finally, the packet reaches the destination node $D$. When the destination node $D$ receives the accumulated RREQ packets, it measures the number of packets received as $P_{rec}$. Then the destination node estimates the success ratio of each node $N_i$ as

$$SR_i = \frac{FC_{N_i}}{P_{rec}}$$

(4.5)

Where, $P_{rec}$ is the number of packets received by the destination node $D$ in the time interval $t_1$. The forward counter values of a node $N_i$, $FC_{N_i}$ are obtained from the corresponding NTIT table of the node. The success ratio $SR_i$ for the node $N_i$ is then added with the Route Reply (RREP) packet. After receiving the RREP packet, the source node reads the success ratio values of all nodes and calculates the trust value.

The source node $S$ checks the success ratio $SR_i$ of all the nodes. For any node $N_k$ if $SR_k < SR_{min}$, where $SR_{min}$ is the minimum threshold value of the success ratio, its trust index is decremented as $T_i = T_i - C$. For all other nodes with $SR_k > SR_{min}$, the trust index values are incremented as $T_i = T_i + C$, where $C$ is the step value. The function for updating the trust index value depends on two parameters, the existing trust value and the estimated trust value. The estimated trust value of the node $k$ is obtained from its trust index (ie) $E_k = T_i$. The following equation is used to update the trust index value for each node encountered in the route.
\[ T_I(E, Te) = (1 - K) * E + K * Te \]  \hspace{1cm} (4.6)

where

\( T_I \): The upgraded trust value

\( Te \): The existing trust value

\( E \): The estimated trust value

\( K \): The constant to express the inflation of trust

### 4.3.3 Estimation of Delivery Probability

The Delivery Probability of the node is calculated using Connectivity \( C_N \), Trust index \( T_I \) and Power level \( P_L \) using the following equation.

\[ DP = W_1 * C_N + W_2 * T_I + W_3 * P_L \]  \hspace{1cm} (4.7)

where, \( W_1, W_2 \) and \( W_3 \) are constants that can be tuned at the run time and the Power level \( P_L \) can be obtained directly from the battery.

### 4.3.4 Forward Error Correction Using LT Codes

Forward Error Correction is performed for reliable messages using Luby Transform (LT) codes. The LT codes are rateless erasure codes with the property that a potentially limitless sequence of encoding symbols can be generated from a given set of source symbols such that the original source symbols can ideally be recovered from any subset of the encoding symbols of size equal to or only slightly larger than the number of source symbols.

The LT codes are rateless since the number of encoding symbols that can be generated from the data is potentially limitless. The exact copy of
the data can be recovered using the decoder from any set of the generated encoding symbols. Thus, the loss model on the erasure channel has nothing to do with the generation of the required encoding symbols. Until the adequate number of generated symbols arrives at the decoder, the generated symbols are sent over the erasure channel to recover the data. The decoder can recover the data from the optimal number of the possible encoding symbols and the LT codes are optimal with respect to any other erasure codes.

In most of the data delivery applications, LT codes can provide better advantages compared with other erasure codes as a minimum number of the encoding symbols is generated by the sender. Similarly, the receiver also requires only a minimum number of the encoded symbols to recover the original data. The applications of the LT codes which ensure resiliency to the network disruptions include the robust distributed storage, the delivery of the streaming content, the delivery of the content to the mobile clients in wireless networks, peer-to-peer applications and the delivery of the content in the multiple paths.

Encoding

The given data of length N is divided into $K = \frac{N}{L}$ input symbols so that every transmitted data is of length L where L is the length of the encoding symbols. There is a key associated with each encoding symbol. The encoder and the decoder apply the same function to the key to produce the degree and set of neighbors of the encoding symbol. The key is selected at random to generate the encoding symbol, and passed to the decoder along with the encoding symbol. Each key generated is larger than the previous key since they are produced by the deterministic process and the same set of random bits is accessed by both the encoder and decoder. Each key is used as the seed to a pseudo-random number generator that uses these random bits to produce the degree and the neighbors of the encoding symbol.
Decoding

The decoder recovers the input symbols repeatedly from the given group of the encoded symbols and an associated degree is computed from the set of neighbors using the following rule. If there is at least one encoding symbol that has exactly one neighbor then the neighbor can be recovered immediately since it is a copy of the encoding symbol. The value of the recovered input symbol is XORed with the other remaining encoded symbols having an input symbol as a neighbor. The recovered input symbol is removed from each of the encoded symbol, as a neighbor and the degree of each such encoded symbol is decreased by one to reflect this removal.

4.3.5 Algorithm Description

Let Delivery Probability be DP, Context Flag, CF, message Urgency, U, Reliability, R, Normal flag, N and Th₁ and Th₂, the minimum and maximum threshold values. Execution of the PECABP protocol is given in Figure 4.3.

1. Each node estimates its DP using the Equation (4.7).

2. Each node exchanges its DP with other nodes which helps it know the neighbor’s information that is stored in it.

3. Before sending the broadcast message, the sender checks the CF.
   i. If CF = U, the message gets forwarded to the nodes with DP > Th₁.
   ii. If CF = R, the message gets forwarded to the nodes with DP > Th₂ and FEC is added with the message for a reliable transmission using LT codes.
   iii. If CF= N, message gets forwarded to the nodes with DP<Th₁.

4. The same process is repeated until the message reaches all the nodes in the network.
Figure 4.3 PECABP Execution

1. Calculate aggregate local mobility
2. Determine Connectivity using Equation 4.4
3. Create NTIT
4. Estimate Trust Index using Equation 4.6
5. Estimate DP using Equation 4.7
6. Each node stores DP of all the neighboring nodes
7. Node checks Context Flag (CF)
8. If CF = U:
   - Yes: Forward message with DP > Th₁
   - No: If CF = R:
     - Yes: Forward message with DP > Th₁
     - No: If CF = N:
       - Yes: Forward message with DP < Th₁
       - No: If broadcasting ends:
         - Yes: Stop
         - No: Apply FEC using LT codes
   - No: If CF = U:
     - Yes: Forward message with DP > Th₁
     - No: If CF = R:
       - Yes: Forward message with DP > Th₁
       - No: If CF = N:
         - Yes: Forward message with DP < Th₁
         - No: If broadcasting ends:
           - Yes: Stop
           - No: Apply FEC using LT codes
4.3.6 Advantages of PECABP

- PECABP protocol delivers the messages based on the priorities such as urgency, reliability and normal.

- Broadcasting is done based on the delivery probability of a node and so, the broadcast redundancy is reduced.

- Since the delivery probability is calculated based on the power level, the protocol is power efficient.

- A reliable message delivery is assured since the delivery probability uses the node’s trust level as one of the parameters.

- The user can control the behavior of the protocol for each message.

4.4 SIMULATION

The proposed PECABP protocol is simulated using Network Simulator-2. The channel capacity of the mobile host is set as a constant value of 2 MBPS. The routing protocol DSDV is used to broadcast the messages synchronously and CAR is used to broadcast the messages asynchronously during network partition.

The DCF of IEEE 802.11 is used for the wireless LANs as the MAC layer protocol that notifies the network layer about link breakage. The number of nodes in the network is varied from 20 to 100. The simulation settings and parameters are summarized in Table 4.1.
### Table 4.1  Simulation Parameters for CABP and PECABP Protocols

<table>
<thead>
<tr>
<th>Simulation Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Size</td>
<td>1000 x1000 Meters</td>
</tr>
<tr>
<td>Number of Nodes</td>
<td>20 to 100</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>2 MBPS</td>
</tr>
<tr>
<td>Routing Protocol</td>
<td>DSDV and CAR</td>
</tr>
<tr>
<td>MESSAGE_PORT</td>
<td>42</td>
</tr>
<tr>
<td>NAM Animation Speed</td>
<td>250 Micro Seconds</td>
</tr>
<tr>
<td>Node Speed</td>
<td>5 Micro Seconds</td>
</tr>
<tr>
<td>MAC</td>
<td>IEEE 802.11</td>
</tr>
<tr>
<td>Radio Range</td>
<td>250 Meters</td>
</tr>
<tr>
<td>Simulation Time</td>
<td>100 Seconds</td>
</tr>
<tr>
<td>Traffic Source</td>
<td>CBR (Constant Bit Rate)</td>
</tr>
<tr>
<td>Broadcast Delay</td>
<td>0.01 Micro Seconds</td>
</tr>
<tr>
<td>Hello Reply Delay</td>
<td>0.01 Micro Seconds</td>
</tr>
<tr>
<td>Pause Time</td>
<td>10,20,30,40,50</td>
</tr>
<tr>
<td>Packet Size</td>
<td>512 Bytes (Max 1500 Bytes)</td>
</tr>
<tr>
<td>Antenna Model</td>
<td>Antenna/OmniAntenna</td>
</tr>
<tr>
<td>Interface queue type</td>
<td>Queue/DropTail/PriQueue</td>
</tr>
<tr>
<td>Max packet in Interface Queue</td>
<td>50</td>
</tr>
<tr>
<td>Mobility Model</td>
<td>Random Way Point</td>
</tr>
<tr>
<td>Transmit Power</td>
<td>0.360 Watts</td>
</tr>
<tr>
<td>Receiving Power</td>
<td>0.395 Watts</td>
</tr>
<tr>
<td>Idle Power</td>
<td>0.335 Watts</td>
</tr>
<tr>
<td>Initial Energy</td>
<td>5.1 Joules</td>
</tr>
</tbody>
</table>
4.4.1 Experimental Setup and Results

The initial phase of the broadcasting with Power Efficient Context Aware Broadcasting Protocol using NAM instance is shown in Figure 4.4. There are 50 nodes in the current simulation and three source nodes namely 4, 44 and 1 are marked in red. The big circle represents the maximum communication range of the source node. Source node 4 broadcasts the urgency message, source node 44, the reliable message and source node 1, the normal message indicated near each node. The three source nodes start broadcasting the messages to all their neighboring nodes.

Figure 4.4 Initial Phase of Broadcasting using PECABP Protocol
Figure 4.5 represents the execution of PECABP protocol that calculates the delivery probability of each node based on the connectivity, power level and trust index. The urgency message from the source node 4 is rebroadcast immediately by the forwarding nodes 25 and 33, whereas the reliable message from the source node 44 is rebroadcast little slowly and normal message from the source node 1 is broadcast more slowly. The nodes receiving the broadcast message are marked in green.

Figure 4.5  Execution of PECABP Broadcasting Protocol
Figure 4.6 represents the functioning of PECABP protocol. The urgency message is broadcast in a faster manner to most of the nodes in the network. The reliable message is broadcast in a little slower manner to many nodes in the network, but still some of the nodes do not receive the reliable message. The normal message from the source node 1 has just started broadcasting.

Figure 4.6 Scenario Representing PECABP Protocol Functioning

Figure 4.7 represents the NAM instance of the final phase of the broadcasting by the PECABP protocol. All the three types of messages, urgency, reliability and normal are broadcast to all the nodes in the network.
4.5 PERFORMANCE METRICS AND ANALYSIS

The Power Efficient Context Aware Broadcasting Protocol is compared with Context Aware Broadcasting Protocol. In the simulation, the mobile nodes move in a 1000 meter x 1000 meter region for the simulation period of 100 seconds. It is assumed that each node moves independently with the same average speed. The following metrics are used to evaluate the performance of these two protocols.
• **Routing Overhead** - It is defined as the ratio of the total number of the routing control packets normalized to the total number of the received data packets.

• **Average End-to-End Delay** - The end-to-end-delay is averaged to the surviving data packets from the sources to the destinations.

• **Packet Delivery Ratio** - It is the ratio between the number of data packets received successfully and the total number of data and control packets sent.

• **Average Energy Consumption** - It is the average energy consumption of all the nodes in performing the operations such as sending, receiving and forwarding the data packets.

### 4.5.1 Based on Pause Time

The pause time of the mobile node is varied as 10, 20, 30, 40 and 50. The speed of the mobile node is set as 5 micro seconds and the simulated traffic is CBR. X-graphs are generated to compare the performance of PECABP and CABP broadcasting protocols based on the above four metrics. Each point in the plot is an average of over hundred simulation runs. The simulation results based on different threshold values are presented to verify and compare the effectiveness of these algorithms.

From Figures 4.8, it is inferred that the Routing Overhead is less for PECABP compared to CABP that considers only the node urgency, not the message reliability. All the messages are treated as equal and the protocol imposes the same overhead on all messages. PECABP protocol considers three different types of messages. Hence, broadcasting the normal and reliable messages requires less overhead because they can tolerate the delay.
Figure 4.8  Comparison between CABP and PECABP Protocols (Pause Time vs. Routing Overhead)

Figure 4.9 represents the comparison of CABP and PECABP protocols on the metric Packet Delivery Ratio. PECBP protocol achieves a better delivery ratio since it considers the node’s connectivity, trust index and power level to calculate the delivery probability. CABP protocol considers the delivery probability based only on message urgency and its packet delivery ratio is much lower compared with the PECABP protocol.
From Figure 4.10, it is observed that the average End-to-End Delay of the proposed PECBP protocol is much less compared with CABP protocol. Since the proposed protocol incorporates the forward error correction techniques for the message encoding, the packet loss is less and so it reduces the unnecessary retransmission for reliable messages. The urgency messages are transmitted immediately and so, end to end delay is less compared with CABP protocol.
Figure 4.10  Comparison between CABP and PECABP Protocols  
(Pause Time vs. Average End-to-End Delay)

Figure 4.11 shows the comparison result of Energy Consumption of 
CABP and PECABP protocols for the pause time 10, 20…50. From the result, 
it is found that PECABP protocol consumes less energy than the CABP 
protocol since it uses the energy efficient routing techniques for broadcasting 
the messages effectively considering the nodes power level for calculating the 
delivery probability of each node.
Figure 4.11  Comparison between CABP and PECABP Protocols
(Pause Time vs. Average Energy Consumption)

4.5.2  Based on Number of Nodes

It is assumed that all nodes have the same transmission range of 250 meters and the network size is varied as 20, 40, 60, 80 and 100 nodes fixing the pause time as 10 seconds. From Figures 4.12, it is inferred that the Routing Overhead is less in PECABP protocol compared with CABP protocol when the number of nodes is increased from 20 to 100.
Figure 4.12 Comparison between CABP and PECABP Protocols (Number of Nodes vs. Routing Overhead)

As the number of nodes increases, a larger number of RREQ and RREP packets are flooded in the network. In PECABP protocol, the trust index is considered to be one of the criteria to determine the delivery probability. Based on the trust index, very few of the neighboring nodes are selected for forwarding the message. Hence, the number of RREQ and RREP packets being transmitted is reduced minimizing the routing overhead in PECABP protocol. PECABP protocol considers delivery probability of each
node to forward the packets, it achieves a better delivery ratio compared with CABP and is shown in Figure 4.13.

![Comparison between CABP and PECABP Protocols](image)

**Figure 4.13** Comparison between CABP and PECABP Protocols  
(Number of Nodes vs. Packet Delivery Ratio)

From Figure 4.14, it is understood that CABP protocol considers all the messages as equal without considering any priority. PECABP differentiates the messages as urgency, reliability and normal. PECABP introduces LT codes for the forward error correction and thus reduces the packet loss and the unwanted retransmission. Hence, Average End-to-End Delay of the proposed PECABP protocol is less compared with the CABP protocol.
Figure 4.14  Comparison between CABP and PECABP Protocols  
(Number of Nodes vs. Average End-to-End Delay)

The comparison between CABP and PECABP protocol on energy consumption is given in Figure 4.15 by varying the number of nodes from 20 to 100. PECABP protocol considers the battery power level of each node to calculate the delivery probability and so more energy efficient compared with CABP broadcasting protocol.
Figure 4.15  Comparison between CABP and PECABP Protocols
(Number of Nodes vs. Average Energy Consumption)

4.6   CONCLUSION

In MANET, the Context-Aware Broadcasting Protocol (CABP) is
designed to provide an efficient broadcasting using the delivery probability
but the protocol suffers an increased average end to end delay and increased
control overhead when the number of nodes or pause time of the node is
increased. A new Power Efficient Context Aware Broadcasting Protocol is
proposed for packet forwarding based on the Delivery Probability (DP).
The main objective of PECABP protocol is to maximize the number of nodes reached and minimize the duration of the broadcasting process and the bandwidth utilization.

Initially the delivery probability of each node is determined based on its connectivity, power level and trust index. Each node maintains a table of its neighbor’s DP. The message header consists of a Context Flag (CF) having three values namely the urgency U, the reliability R and the normal N. The urgency messages are delivered in a short time with tolerable packet losses. The messages having the flag value reliability are delivered without any data loss but with a tolerable delay. To ensure lossless delivery, the forward error correction is performed on the reliability messages using LT codes. The other messages are treated as normal that can tolerate both the delay and packet loss. The delivery probabilities of the nodes are updated periodically to maintain the consistency. From the simulation results, it is proved that PECABP protocol is more power efficient and reliable, reduces the broadcast redundancy better compared with CABP broadcasting protocol.