Energy minimization Using binary to ternary conversion along with ACO

Prof. Atul Pawar, Tejas Patil, Pavan Musale, Jaystu Wavare, Rohit Madiwale
Tdpatif983@gmail.com, pavannusale06@gmail.com, javastu77@gmail.com
Department of Computer Engineering, PCCOE Pune

Abstract
Today Wireless Sensor Networks (WSNs) are utilized as a part of various ventures. Wireless sensor networks as a rule require ease gadgets and low power activities. WSN involve innumerable sensor nodes. Amplifying node or network lifetime is essential goal in WSN on the grounds that it is exceptionally hard to charge or supplant depleted batteries. In this way energy powerful correspondence is exceptionally critical part in WSN to decrease the gadget reviving cycle period and henceforth give network to longer span. So there are such a large number of plan proposed like energy productive correspondence coding technique for WSN which depends on the ternary number framework encoding of information. A proficient calculation for transformation from two fold to ternary and the other way around is utilized that does not include any division or duplication but rather just expansion. Insect Colony Optimization Algorithm is additionally used to discover most limited way between source node and goal node, which decreases energy minimization.

Introduction
Wireless sensor networks (WSNs) consistently utilize uncommonly imperativeness obliged, ease sensor devices that are passed on (every now and again in an improvised path) in districts that are hard to get to and with by zero network system. As a rule, such battery controlled sensor devices are required to work over postponed time spans. Correspondence being an imperative wellspring of energy drain in such networks, imperativeness proficient correspondence traditions that can be executed with low gear and programming cost/multifaceted design are in this way of principal importance in WSNs to lessen the contraption stimulating cycle periods and consequently offer accessibility to longer traverses at a broaden. For all intents and purposes, most existing transmission designs not simply utilize non-zero voltage levels for both 0 and 1 so as to perceive a clam and a clamoring channel, they in like manner keep both the transmitter and the beneficiary traded on for the entire traverse of the transmission of a data plot. Correspondence frameworks that require imperativeness utilization for transmitting both 0 and 1 bit regards are known as energy based transmission (EbT) plans.

By the day's end, if the essentialness required per bit transmitted is e , the total imperativeness ate up to transmit a n-bit data would be n e b . Most force analyse endeavours on diminishing essentialness use have focussed on the MAC layer design enhancing data transmissions by b decreasing effects and retransmissions and through astute decision of ways or uncommon models for sending data. In each and every such arrangement, the fundamental correspondence procedure of sending a string of combined bits is essentialness based transmission. As opposed to EbT based correspondence plots, another correspondence method
called Communication through Silence (CtS) was suggested that incorporates the use of quiet periods rather than imperativeness based transmissions. CtS, in any case shortcoming of being exponential in correspondence time. An alternative framework, called VariableBase Tacit Communication (VarBaTaC) was prescribed that uses a variable radix-based information coding joined with CtS for correspondence. Regardless, for a n-bit combined string, the length of transmission is overall significantly longer than n. Some maker never talk about the measure of energy saved by CtS and VarBaTaC for uproarious channels and thinking about real contraption characteristics. Acquiring from the thoughts of CtS and VarBaTaC, proposed another imperativeness efficient contrive called RBNSiZeComm for wireless sensor networks that recodes a combined coded data using a monotonous radix based number depiction and after that uses silent periods to pass on the bit estimation of 0, the overabundance parallel number system (RBNS) that uses the digits from the set -1,0,1 to address a number with radix 2, it is possible to essentially decrease the amount of non-zero digits that ought to be transmitted.

Proposed System

![Proposed System Diagram]

Figure a Proposed System for future implementation

The aim of generating energy saving simultaneously at the transmitter and receiver side. For that system will implement using a new communication scheme based on recoding data from binary to the ternary radix and the silent symbol strategy and Ant Colony Optimization Technique (ACO).

A new low energy communication scheme that can generate energy savings simultaneously at the transmitter and the receiver, unlike the RBNSiZeComm protocol. An efficient algorithm involving only addition (and no multiplication or division) for conversion from binary to ternary and vice versa is used in order to keep the energy consumed for the radix conversion process low at both the transmitter and the receiver. Coupled with the ACO Algorithm will associate finding shortest distance between numbers of nodes.

ACO Algorithm steps

1. Initialization:
   Discretize the storage volume into several classes for each time period
   Initialize pheromone trail $\tau_i$, and other parameters (e.g., $\alpha$, $\beta$, $p$ values)

2. Solution construction:
   Position each ant in a starting node (i.e., Random distribute the starting period and class of initial reservoir volume for each ant)
   For each ant $k$ do
     Repeat
     Compute heuristic information $\tau_k$
     Choose next node by applying the state transition rule given by equations (1) & (2)
     Until every ant has built a solution
     Compute fitness value given by equation (14)
   End for

3. Trail update:
   Update the best solution
   Apply offline pheromone trail update
   For iteration best ant $\text{global best ant move (i,j)}$ do
     Compute $\Delta\tau_i$
     Update the trail value by means of equations (4)
   End for

4. Terminating condition:
   If (end condition = true)
     then print the best result so far
   else go to step 2

Let a given binary message $B$ be represented by an $n$-bit binary string $b_n b_{n-1} b_{n-2} \ldots b_2 b_1 b_0$. Let $T$ be its equivalent $m$-digit ternary representation given by $T = t_m t_{m-1} t_{m-2} \ldots t_2 t_1 t_0$. We assume that $n$ is even, otherwise we pad the message with a 0 bit at the leftmost (msb) position.
For conversion from binary to ternary, we successively scan every two bits of the binary message starting from its msb position and then convert the leftmost part of the binary message (starting from its msb to the currently scanned bit position) to its equivalent ternary representation. To do this, we replace the currently scanned two bits (bit-pair) by either a single ternary digit as shown in below table.

<table>
<thead>
<tr>
<th>Binary bit pair</th>
<th>Ternary bits (radix-3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(00)</td>
<td>(0)3</td>
</tr>
<tr>
<td>(01)</td>
<td>(1)3</td>
</tr>
<tr>
<td>(10)</td>
<td>(2)3</td>
</tr>
<tr>
<td>(11)</td>
<td>(10)3</td>
</tr>
</tbody>
</table>

For conversion from binary to ternary, we successively scan every two bits of the binary message starting from its msb position and then convert the leftmost part of the binary message (starting from its msb to the currently scanned bit position) to its equivalent ternary representation. To do this, we replace the currently scanned two bits (bit-pair) by either a single ternary digit (0, 1 or 2 depending on whether the binary bit-pair is 00, 01 or 10), or by two ternary digits (10) (if the binary bit-pair is 11), and then add it to four times the so far obtained ternary number T from all the previous bit positions (on the left of this bit-pair). Note that this addition will be in ternary number system, and the multiplying factor four is to adjust the weight of T with respect to the currently scanned bit-pair. Also, since this addition will be in ternary number system, the weight of four can be assigned to T by adding the two ternary numbers T0 and 0T. Thus, the equivalent ternary representation of the part of the binary message from its msb to the currently scanned bit-pair will then be obtained as T00T x, where represents a ternary addition operator, and x is the ternary number equal to 0, 1, or 2 (when the scanned bit-pair is 00, 01 or 10, respectively), or equal to (10) (when the scanned bi-pair is 11). In a similar manner, we can reconvert the received m- digit ternary message to its equivalent binary form by using algorithm Ternary2Binary, where again we scan the given ternary number from its most significant (leftmost) digit position in a digit by digit manner, and convert the part of the so far scanned ternary digits to its equivalent binary representation. In the algorithm, we denote a ternary digit by and the converted binary number by B. As in the previous Binary2Ternary algorithm, we need to adjust the weight of the binary representation of the previously scanned ternary digits, by attaching a weight of three to it, with respect to the currently scanned ternary digit, and this is achieved by adding B0, 0B and the equivalent binary representation of the currently scanned ternary digit (note that, this time it is a binary addition).
procedure BINARY2TERNARY(IN bit vector \( \tilde{b} \), integer \( n \), OUT ternary number \( T \))
/* Initialization */
if \( b_{n-1}b_{n-2} = 00 \) then
\( T \leftarrow 0; \)
end if
if \( b_{n-1}b_{n-2} = 01 \) then
\( T \leftarrow 1; \)
end if
if \( b_{n-1}b_{n-2} = 10 \) then
\( T \leftarrow 2; \)
end if
if \( b_{n-1}b_{n-2} = 11 \) then
\( T \leftarrow 10; \)
end if

for \( i = n - 3 \) down to 1 step -2 do
if \( b_{i+1}b_i = 00 \) then
\( T \leftarrow T \oplus 00; \) /* this is a ternary addition */
end if
if \( b_{i+1}b_i = 01 \) then
\( T \leftarrow T \oplus 01; \) /* ternary addition */
end if
if \( b_{i+1}b_i = 10 \) then
\( T \leftarrow T \oplus 02; \) /* ternary addition */
end if
if \( b_{i+1}b_i = 11 \) then
\( T \leftarrow T \oplus 10; \) /* ternary addition */
end if
end for
end procedure

procedure TERNARY2BINARY(IN ternary number \( T \), integer \( m \), OUT binary number \( B \))
/* Initialization */
if \( t_{m-1} = 0 \) then
\( B \leftarrow 00; \)
end if
if \( t_{m-1} = 1 \) then
\( B \leftarrow 01; \)
end if
if \( t_{m-1} = 2 \) then
\( B \leftarrow 10; \)
end if

for \( i = m - 2 \) to 0 step -1 do
if \( t_i = 0 \) then
\( B \leftarrow B0 + 0B; \) /* this is a binary addition */
end if
if \( t_i = 1 \) then
\( B \leftarrow B0 + 0B + 1; \) /* binary addition */
end if
if \( t_i = 2 \) then
\( B \leftarrow B0 + 0B + 10; \) /* binary addition */
end if
end for
end procedure
AODV routing protocol

It provides efficient communication among nodes in Ad hoc network. The routing protocol is subject to some issues and challenges due to wireless mode of transmission. For efficient design of routing protocol one has to consider design issues as discussed below.

Resource estimation: During route establishment, estimation of available bandwidth to a node or link and delay must be done. Bandwidth available to a link or node dynamically vary which in turn affects by the traffic of its neighbouring nodes.

Route discovery: Route discovery is based on choice of routing. Proactive routing leads to increase overhead with less delay. Reactive routing at the expense of more delay reduces routing overhead. To support QoS aware routing it is desirable to achieve routing with less overhead and latency. This is achieved by AODV reactive routing protocol.

Resource reservation: The fair distribution of resources among hosts in Ad hoc networks is one of the challenging issue. To overcome this problem resource reservation scheme can be used to set and maintain QoS-aware routing.

Route maintenance: In MANETs, mobility of nodes causes topology to change frequently, making it difficult to meet the QoS constraints. Biggest design issue is incorporating fast route maintenance scheme which discover a route break up.

Route selection: As topology change causes route failures and affects the end-to-end QoS, there is need to avoid route failure. For efficient design of QoS aware routing the route with maximum available bandwidth has to be considered.

Route failure notification: Routing protocol must provide information about remaining bandwidth or estimation of route delay through feedback to the application.

Working of AODV

Only when a mobile terminal has packets to send to a destination does it need to discover and maintain a route to that destination terminal. In AODV, each terminal contains a route table for a destination. A route table stores the following information: destination address and its sequence number, active neighbours for the route, hop count to the destination, and expiration time for the table. The expiration time is updated each time the route is used. If this route has not been used for a specified period of time, it is discarded.

During route discovery process, it broadcasts RREQ to all its neighbours for specified destination. The RREQ is then flooded throughout the network at once. An intermediate node on receiving a RREQ, checks its routing table, to find route to destination. If destination is found then it sends RREP to source by setting up a reverse route path to source node in its route table. If not found, intermediate node rebroadcast RREQ to its neighbour nodes. The intermediate node ignores RREQ if it has processed already [11]. The process is repeated till destination node is found and on reaching destination node with RREQ, the destination node unicast RREP (route reply) to source node.

Algorithm: AODV routing algorithm

Step1: Start
Step2: Neighbour node identification
Step 3: Find route to destination
Step 3.1: Broadcast RREQ to neighbour nodes
Step 3.2: If route present, neighbour sends RREP to source
Step 3.3: If no route, neighbour initiates route discovery
Step 4: For route discovery by intermediate node, go to step 3.
Step 5: Data transmission phase
Step 5.1: Transmit data to next hop neighbour
Step 5.2: If route break identified
Step 5.2.1: Notify the source with an RERR.
Step 5.2.2: Go to step 3
Step 6: Performance evaluation
Step 7: End

Result
### Table 1: Energy Consumed By Nodes

<table>
<thead>
<tr>
<th>No. of node</th>
<th>Energy consumed by Aodv</th>
<th>Energy consumed by ACO+BT/TB</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>18</td>
<td>16</td>
</tr>
<tr>
<td>150</td>
<td>38</td>
<td>33</td>
</tr>
<tr>
<td>200</td>
<td>56</td>
<td>51</td>
</tr>
<tr>
<td>250</td>
<td>74</td>
<td>66</td>
</tr>
<tr>
<td>300</td>
<td>86</td>
<td>78</td>
</tr>
<tr>
<td>350</td>
<td>102</td>
<td>91</td>
</tr>
<tr>
<td>400</td>
<td>110</td>
<td>103</td>
</tr>
</tbody>
</table>

### Conclusion

In this paper Aco algorithm and Binary to ternary and ternary to Binary algorithm have been implemented. Paper focused on energy consumption of nodes. Result has shown clearly using BT/TB algorithm with ACO is better than aodv.

### References


