LOCATION BASED ENERGY EFFICIENT SCHEME (LNE^2S) FOR MOBILE AD HOC NETWORKS

Subhankar Ghosh*, Anuradha Banerjee†
*Department of Computer Science and Engineering,
Regent Education and Research Foundation, West Bengal, India
Email: redhatsubha@gmail.com
†Department of Computer Applications,
Kalyani Govt. Engg. College, West Bengal, India
Email: anuradha79bn@gmail.com

***Abstract*—Based on the topology of underlying ad hoc network, an energy efficient scheme is proposed that takes care of comparative cost in individual links. Within single hop downlink neighbourhood of a node, multiple paths (single or multi-hop) may exist to various downlink neighbours. Depending upon energy consumption in each of them, LNE^2S selects the optimum one. Please note that, by the cost of communication from node n_i to n_j, LNE^2S understands the sum of costs of communication from n_i to n_j and n_j to n_i, because acknowledgements are indispensable for successful completion of a communication session. This is a very unique feature of LNE^2S. Moreover, an weight based sleeping strategy is also proposed where the best alternative of a node n_i bridges the gap between communicating uplink and downlink neighbours of n_i, where n_i goes to sleep.**

***Keywords*—Ad hoc network, downlink neighbor, energy efficiency, battery powered, energy-oriented link life, velocity oriented link life.**

1. INTRODUCTION

Ad hoc networks are self organized and consist of only some nodes that move freely with arbitrary velocity and direction. They do not require help of any pre-existing infrastructure or centralized administration. These networks are very helpful in emergency situations like war, natural disasters like flood, earthquake etc. where traditional wired networks fail to work. However, ad hoc networks suffer from challenges like unpredictable mobility, security, limited battery power and bandwidth [1]. Researchers are working hard to address these issues [2]. Since nodes in ad hoc networks are battery powered, energy efficiency is a matter of great concern for longevity and capability of the network. According to [3], at least 40% of initial or maximum battery power is required to remain in operable condition; 40% - 60% is satisfactory, 60% - 80% is good whereas the next higher range i.e. 80% - 100% is considered to be more than sufficient. Therefore it must be appreciated that battery power is a scarce resource and if in a multi-hop path, routers are not equipped with sufficient residual battery power, it may seriously affect percentage of live nodes in the network. Reason is that, as soon as a router n_j in an active communication path, runs out of battery power, the link from its predecessor n_i to n_j breaks. In order to repair the broken link n_i soon broadcasts route-request (RREQ) messages in the network. Routers that forward those messages will consume additional energy. If a large number of links in the network, break, then a lot of energy of various nodes will be simply wasted in forwarding RREQ packets. Some nodes may die and network may get partitioned too. All these would not have happened, if routers in active communication paths were equipped with sufficient energy. Therefore, preserving energy is very crucial for ad hoc networks.

In recent years, many techniques have been proposed to conserve energy in ad hoc networks. Some of them are, adjusting transmission power depending upon the distance between sender and receiver in a link, allowing nodes to go to sleep provided a suitable alternative is there. Our present contribution LNE^2S concentrates on finding optimum path to one hop downlink neighbors of a node n_i. Transmission power is also adjusted depending upon location of each individual downlink neighbor. Acknowledgements have an important role to play too. Reducing cost of communication from n_i to n_k requires reduction in cost of transmitting acknowledgements from n_k to n_i, too. Routers on the verge of exhaustion are allowed to go to sleep provided suitable replacements are available. This allows preserving energy without compromising with network throughput.

Rest of the paper is organized as follows. In section 2, some state of the art energy preservation techniques are discussed. Proposed technique is described in section 3. Section 4 illustrates LNE^2S with some examples. Sleeping strategy is described in section 5. Simulation results appear in section 6 while section 7 concludes the paper.

2. RELATED WORK

Energy conservation techniques in ad hoc networks apply mainly the following two approaches -
i) Adjusting transmission power
ii) Putting as many nodes as possible in sleep state

Radio transceivers are switched off in sleep state. Over-exhausted nodes are allowed to go to sleep for a pre-defined time duration after which they wake up and change state from sleep to idle. In idle state, a node waits for incoming network traffic. As soon as some packet comes for forwarding, the idle node changes state to active. Consumption of energy in idle state is lesser than consumption of energy in active state. Consumption of energy is zero in sleep state [3], [4], [5], [6], [7], [8]. Least energy is consumed by network nodes provided all nodes are switched off, but that will completely disrupt all communication activities. Hence, a trade-off has to be there between energy preservation and network throughput [9], [10], [11], [12].

Adjusting transmission power is explained below in figure 1, n_j is a 1-hop downlink neighbor of n_i. Maximum transmission power and radio range of n_i are denoted as $P_{max}(i)$ and R_i respectively. Also assume that $dist_{ij}(t)$ is the Cartesian distance between n_i and n_j at time t.

For the signal to travel the distance $dist_{ij}(t)$, required transmission power $P_j(i)$ is formulated below:

$P_j(i) = P_{max}(i)\left(\frac{dist_{ij}(t)}{R}\right)^q$

where q can take values 2, 3 or 4 depending upon the medium of communication [9], [10], [13], [14]. For simplicity of computation, in LNE^2S, we have taken q=2. Topology control techniques require computing Cartesian distance between two nodes. Hence, all nodes need to be equipped with global positioning system (GPS). Geographic adaptive fidelity (GAF) and span save energy through maintenance of connected dominating set (CDS) of a network. In, GAF, if two nodes n_j and n_i have same set of uplink and downlink neighbours, then n_i and n_j are termed as redundant. If n_i needs to go to sleep, then n_j can bridge the gap between uplink and downlink neighbours of n_i. Hence, n_i can take a map without disrupting message forwarding in the network. Similarly, when n_j will feel exhausted and need to go to sleep, n_i will take responsibility of its forwarding tasks. In SPAN, a set of nodes always remain alive; they take the responsibility of communication, when the set gets exhausted, it is replaced by a new set of nodes. In [14], a power saving tree is constructed for distribution of power control (ANTC), sending nodes adaptively change transmission power of nodes based on location information of itself and the 1-hop downlink neighbour with which it wants to communicate. A backbone is selected by the nodes that guarantee a hierarchical topological structure but there is no provision to go to sleep [15], [16], [17], [18], [19], [20]. All nodes have to continuously remain awake. Apparently it seems that this will improve network throughput, but actually throughput will decrease due to high degree of link breakage and possible network partitions. In [21], a topology control technique named coopsink is proposed where nodes cooperate with each other through decode and forward strategy [22]. This reduces energy consumption along with suggesting an energy-efficient route to destination. A routing protocol [23] is proposed where if a router n_j discovers that the strength of acknowledgement message sent by a 1-hop downlink neighbour n_k is very weak, then n_i understands that n_k is on the verge of getting out of radio-circle of n_i. So, n_i starts communicating with destination through some other path (that does not contain n_k) discovered during route discovery session itself. Please note that, unlike other protocols top three choices of communication routes are stored in source as well as routers, instead of one optimum choice in conventional protocols. A distributed energy efficient multicast algorithm is proposed in [24] where priority is assigned to nodes in a multicast tree by considering both hop count (distance of that node from root of the tree) and energy consumed across the link [25], [26], [27].

3. PROPOSED WORK (LNE^2S)

In this section, we propose an energy presentation technique that is threefold. It's characteristics are as follows:-

i) It controls transmission power of a node n_i based on the distance each of it's communicating downlink neighbour n_j from n_i.

ii) Optimum route to each of the downlink neighbours is elected. It considers transmission of data packet from a node n_i to a downlink neighbour n_j and also transmission of acknowledgements (ACK) from n_j to n_i, may be in single or multi-hop paths.

This is an unique feature of LNE^2S.

iii) An weight based sleeping strategy is also embedded in LNE^2S that allows some specific downlink neighbours of a node, to go to sleep provided some suitable alternative is available. The sleep duration z is pre-defined.

3.1 Network Model

The network is modelled as a graph G = (V, E) where V represents the set of nodes and E represents the set of edges. N number of heterogeneous nodes are randomly deployed in the network. Each node is identified through an unique identification number known as node-id, and is equipped with GPS as well as antenna. Therefore they are location aware and can very easily calculate their geographical distance from another node. Radio-range R_i of a node n_i is defined to be an abstract geographical circle around the node within which n_i can directly send signals.

After that signal crosses distance R_i, it fades away. All nodes n_j residing within the radio-range of n_i, are termed as 1-hop downlink neighbour or simply neighbour of n_i. Assuming $(x_i(t),\ y_i(t))$ be the latitude and longitude of n_i at time t, condition (1) has to be satisfied by n_j in order to be a 1-hop downlink neighbour of n_i at time t.

$$dist_{ij}(t) \leq R_i \tag{1}$$

In (1), $dist_{ij}(t)$ is the cartesian distance between n_i and n_j at time t. It is formulated in (2).

$$dist_{ij}(t) = \sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2} \tag{2}$$

The set of all 1-hop downlink neighbours of n_i at time t, is denoted as $D_i(t)$.A node n_k will be a 2-hop downlink neighbour of n_i at time t provided n_k is not already a 1-hop downlink neighbour of n_i at time t and 1-hop downlink neighbour of some node $n_j \epsilon D_i(t)$. This is mathematically expressed in conditions (3) and (4).

$$dist_{ik}(t) > R_i \tag{3}$$

$\exists$ n_j ϵ $D_i(t)$ s.t,

$$dist_{jk}(t) \leq R_j \tag{4}$$

Each node n_i in the network periodically broadcasts HELLO message at regular intervals with it's maximum transmission power $P_{max}(i)$. Please note that transmission power control is not possible for HELLO messages because HELLO and other broadcast messages are supposed to cover the entire radio-circle of a node. After receiving HELLO message from n_i, each downlink neighbour n_j of n_i reply with ACK. Assume that n_j replies with ACK at time $t^{'}$ to acknowledge the HELLO message sent by n_i at time t.

Components of HELLO message sent by n_i at time t are as follows:-

i) sender ID i.e. n_i
ii) Position at time t i.e. $(x_i(t), y_i(t))$
iii) radio-range or R_i
iv) Maximum transmission power or $P_{max}(i)$
v) Average velocity at time t i.e. $v_i(t)$
vi) Number of known downlink neighbours current timestamp

Components of ACK message sent by n_j at time $t^{'}$ are as follows:-

i) sender ID or n_j
ii) ID sequence i.e. sequence of nodes from sender n_j to n_i through which ACK arrived at n_i
iii) Position at time $t^{'}$ i.e. ($x_i(t^{'})$,$y_i(t^{'})$)
$iv)$ radio-range R_j
v) Maximum transmission power $P_{max}(j)$
vi) Average velocity at time $t^{'}$ i.e. $v_j(t^{'})$
$vii)$ Call departure rate at time $t^{'}$ i.e. $dep_j(t^{'})$
$viii)$ Residual energy at time $t^{'}$ or $res-eng_j(t^{'})$
$ix)$ Maximum energy or max-eng_j
x) Number of known downlink neighbour
xi) Current timestamp

Based on these information, one efficient path table (EPT) is constructed with the following attributes:-

i) Downlink neighbours id
ii) Last known location in terms of x and y coordinates
iii) Radio-range
iv) Maximum transmission range
v) maximum energy
vi) average velocity in m/s
vii) residual energy
viii) rate of energy consumption
ix) Most efficient path (MEP) to the node
x) MEA of that path
xi) Energy cost of MEP
xii) Energy cost of MEA
xiii) Overall energy cost of the MEP

Before computation of MEA and MEP, those fields contain NYC i.e. not yet computed.

During link selection in LNE^2S, energy-efficient paths are computed from n_i to each of it's downlink neighbours. Corresponding traversal of c has to be energy efficient as well. Please note that, link breakage during transmission of acknowledgement will also require injection of route-request (RREQ) packets in the network. This is equally dangerous and energy consuming like link breakage during data transmission. Therefore, energy consumption in one particular data path (data path is the path through which data packet flows) should be added to the energy consumption in corresponding acknowledgement path in order to get an estimate of overall energy consumption in a communication session. For one particular data path, there may exist more than one acknowledgement path or ACK path. Among all those possible ACK paths, the most eligible one (Most Eligible Acknowledgement or MEA) is elected by LNE^2S to acknowledge data packets. This is a completely novel feature of LNE^2S that enforces the importance of energy-efficient acknowledgements. LNE^2S emphasizes the fact that even if a data packet reaches it's destination, communication session can not complete unless acknowledgement reaches all senders in the data path. Finding energy efficient paths in the network, give rise to a new network topology $G^{'}$ = (V, $E^{'}$) where only power conserving links are present.

Below in section 3.2 we describe the importance of ACK paths corresponding to various data paths. As far as power saving techniques are concerned, in section 3.3 we present transmission power control techniques, then optimum path selection in section 3.4 and obtaining $G^{'}$ (new energy optimized network graph topology) from G through an example (old network topology) in section 3.5.

3.2 WHY ACKS ARE IMPORTANT ?

In LNE^2S, whenever a source node n_s communicates with a destination n_d, n_s injects RREQ packets into the network specifying it's own id (n_s), own location ($x_s(t)$, $y_s(t)$) at current time t, average velocity $v_s(t)$ at t, residual energy at t ($res-eng_s(t)$), maximum energy ($max-eng_s$) and total number of packets PKTs to be delivered to n_d in the current session. After route selection by the underlying routing protocol, route-reply packets (RREP) are sent from n_d to n_s mentioning the optimal path in terms of sequence of node identifiers and their locations. If the link $n_i \rightarrow n_j$ ($n_j \in D_i(t)$) is mentioned in that optimal path, then data packets from n_s will come to n_i which n_i will have to deliver to n_j through an energy optimized path. Selection of this energy optimized path from n_i to n_j is within the jurisdiction of LEES. Please remember that communication from n_i to n_j will remain incomplete unless ACK arrives from n_j to n_i as well as all predecessors of n_j. For example consider figures 2 and 3 where n_i can send data to n_j through two paths. One is n_i →n$_j$ direct and the other one is n_i →n$_X$ →n$_V$ →n$_j$. Suppose, in terms of residual energy and relative velocity, the path n_i →n$_j$ is not expected to survive till end of the just established communication session which the route n_i →n$_X$ →n$_V$ →n$_j$ can do.

On the other hand, as far as acknowledging data packets of n_i are concerned, without any loss of generality, we assume that both the paths n_j →n$_P$ →n$_i$ and n_j →n$_Q$ →n$_i$ can survive till the end of communication session. In that case, since LEES is concerned more with energy efficiency, it will elect the one that consumes lesser energy. Again, without any loss of generality we assume that n_j →n$_P$ →n$_i$ consumes lesser energy than n_j →n$_Q$ →n$_i$. In that case, acknowledgements to data packets shall have to traverse the path n_j →n$_P$ →n$_i$ →n$_X$ →n$_V$ because it is important to inform n_X and n_V that the data packet they forwarded could properly reach n_j. On contrary to it, if we had chosen the path n_i →n$_j$ for forwarding data packets, then path acknowledgement would have been n_j →n$_P$ →n$_i$.

3.3 Transmission Power Control Technique

Considering a link from n_i to n_j at time t, let $P_{min-recv}(j)$ be minimum receive power of node n_j. Then required transmission power $P_{req-trans}(i,t)$ of n_i at time t, is mathematically expressed in (5) and (6).

$$P_{req-trans}(i,t) = \frac{P_{min-recv}(j)\{dist_{ij}(t)\}^q}{C} \quad (5)$$

where C is a constant and q can take value 2,3 or 4. Here, for simplicity, we have taken q=2.

$$P_{req-trans}(i,t) = \frac{P_{min-recv}(j)\{1 + dist_{ij}(t)\}^q}{C} \quad (6)$$

3.4 Optimum Link Selection to 1-hop Neighbour

In order to discover the energy optimized path from n_i to n_j, we need to compute approximate life span of all involved links. The procedure to compute life duration of a link from the perspective of energy and relative velocity, is described below in subsection 3.4.1. In 3.4.2, we illustrate how weight of each data and ack path can be calculated. MEA or most eligible acknowledgement corresponding to energy single data path and based on that, selection of optimized path is presented in section 3.4.3, section 4 gives an example of computation of energy optimized path from n_i to one of it's downlink neighbours n_j.

3.4.1 Link Life Estimation

A link n_i →n$_j$ survives till both n_i and n_j are alive i.e. are equipped with sufficient battery power and n_j resides within the radio-range of n_i. Hence computation of link life span in LEES is divided into computation of energy-oriented link life and velocity-oriented link life.

3.4.1.1 Energy Oriented Approximate Link Life

Considering the link n_i →n$_j$, rate of call departure in n_i at current time t is $dep_i(t)$ i.e. $dep_i(t)$ is the number of packets forwarded by n_i in unit time, at time t. Please assume that n_i consumes $uni-eng_i$ amount of energy to forward one packet. Hence, rate of energy consumption in n_i at time t is given by $rate - eng_i(t)$, which is defined in (7).

$$rate - eng_i(t) = (uni - eng_i \times dep_i(t)) \quad (7)$$

Residual energy of n_i at t is $res - eng_i(t)$ and maximum energy of the same is $max - eng_i$. So, residual operational life span of n_i at time t, is denoted by ls(i, t) and defined as,

$$ls(i,t) = \frac{res - eng_i(t) - 0.4 \times max - eng_i}{uni - eng_i \times dep_i(t)} \quad (8)$$

The formulation in (8) is based on the observation that a node remains operational when it is equipped with at least 40% of it's maximum battery power. So, we always want to preserve at least (0.4 x $max-eng_i$) amount of charge for n_i. If n_i wants to take part in communication sessions, it must have energy higher than (0.4 x $max - eng_i$). Therefore, residual operational life span of n_i at time t, is denoted by ls(i, t) and defined as,

$$ls(i,t) = \frac{res - eng_i(t) - 0.4 \times max - eng_i}{uni - eng_i \times dep_i(t)} \quad (9)$$

Similarly ls(j, t) can also be computed. Energy oriented approximate lifespan eng-span(i, j, t) of the link from n_i to n_j at time t is given by (9).

$$eng - span(i,j,t) = min(ls(i,t), ls(j,t)) \quad (10)$$

3.4.1.2 Velocity Oriented Life Span

Let $v_i(t)$ and $v_j(t)$ denote velocity of n_i and n_j, respectively, at time t. If direction of movement of n_i makes an angle θ with direction of movement of n_j, as shown in fig 4, then resultant velocity in horizontal direction will be (V$_i(t)$cosθ - $V_j(t)$) and the same in vertical direction will be $V_i(t)sin\theta$. Combining these two, overall resultant velocity $reslv_{i,j}(t)$ of n_i w.r.t n_j at time t, is formulated in (11).

$$reslv_{i,j}(t) = \sqrt{(v_i(t)cos\theta - v_j(t))^2 + (v_i(t)sin\theta)^2} \quad (11)$$

i.e.

$$reslv_{i,j}(t) = \sqrt{v_i(t)^2 + v_j(t))^2 - 2v_i(t)v_j(t)cos\theta} \quad (12)$$

Depending upon whether n_i and n_j move towards or away from each other and whether n_j is situated at left or right side of n_i, the distance $D^{'}_{ij}(t)$ yet to be travelled by n_j to get out of radio-range of n_i, is computed below in table 5. In this table, $V_i(t)cos\theta \rightarrow$ L (or R) means direction of $V_i(t)cos\theta$ is to the left (or right).

Similarly $V_j(t) \rightarrow$ L (or R) means direction of $V_j(t)$ is to the left (or right) $dist_{ij}(t)$ denotes distance between n_i and n_j at current time t. RHV and RVV are short forms of resultant horizontal velocity and resultant vertical velocity.

Let vel-span(i,j,t) denote velocity oriented life span of the link from n_i to n_j at time t.
It is mathematically modeled in (13).

$$vel - span(i,j,t) = \frac{D^{'}_{ij}(t)}{reslv_{ij}(t)} \tag{13}$$

Case-1

Direction of movement of n_j is -ve i.e. n_j is going in just opposite direction of n_i. In this case n_j will have to cross the distance (R_i - $distance_{ij}(q(t))$) with average relative velocity $ev_j(i,t)$. This is shown in fig 4.

Residual life span of the link n_i →n$_j$ at time t, is denoted as vel–span(i, j, t) and is shown in (12a).

Case-2

Direction of movement of n_j is +ve i.e. n_j is coming close to n_i.

Residual life span of the link n_i →n$_j$ at time t is denoted as vel–span(i, j, t) and is shown in (12b).

$$vel - span(i,j,t) = \left\{ \frac{R_i - distance_{ij}(q(t))}{ev_j(i,t)} \right\} \tag{14a}$$

$$vel - span(i,j,t) = \left\{ \frac{R_i + distance_{ij}(q(t))}{ev_j(i,t)} \right\} \tag{14b}$$

Lifetime span(i, j, t) of the link n_i →n$_j$ at time t, is shown in (13).

$$span(i,j,t) = min(eng - span(i,j,t), vel - span(i,j,t)) \tag{15}$$

3.4.2 Route Efficiency Computation

Election of the optimal route depends on the following factors:-
i) expected life span ii) duration of the communication session iii) energy consumed ordinarily

Consider the figures 3a and 3b that show different data and ack path options corresponding to the mode n_i and it's active successor n_j.

Data path option 1

The scenario of forwarding data and ack packets in order to complete communication along the link n_i →n$_j$ is shown below :
data path : $n_i \xrightarrow{t_1}$n$_j$ (started at time t)

corresponding acks:

1. $n_j \xrightarrow{t^{'}_1}n_i$ (started at time $t^{'}$)
2. $n_j \xrightarrow{t^{'}_2}n_x$ $\xrightarrow{t^{'}_3}$n$_y$ $\xrightarrow{t^{'}_4}$n$_i$ (started at time $t^{'}$)

For each link above the → sign, some time duration is specified. For e.g, $n_j \xrightarrow{t^{'}_2}n_x$; it shows n_j needs $t^{'}_2$ time to send ack to n_x. n_x receives it at time $t^{'}$ + $t^{'}_2$. Energy consumed per packet in this session by data path option 1, initiated by n_i at time t, is computed based on lifetime of the link n_i →n$_j$ at time t i.e. span(i,j,t) and estimated time of completion of the communication session. Let n_i forwarded RREQ at time tm_{RREQ} and as per RREP sent by the destination, RREQ reached the destination at time tm_{CMP}. Therefore, approximate delay $del_{s,d}(i)$ in the communication path as observed by n_i is given by (14).

$$del_{s,d}(i) = tm_{CMP} - tm_{RREQ} \tag{16}$$

Let pkts be the number of packets to be forwarded from n_s to n_d and intr be the interval between two consecutive packets. So, time required for the communication to be complete for n_i, is denoted by $cm_{s,d}(i)$ and defined in (15).

$$cm_{s,d}(i) = del_{s,d}(i) + (pkts - 1)intr \tag{17}$$

Without any loss of generality, we assume that
i) span (i, j, t) ≥cm$_{s,d}(i)$
ii) span (j, i, t) ≤cm$_{s,d}(i)$
iii) $f - span_{t^{'},t^{'}_2,t^{'}_3}$(j,x,y,i) ≥cm$_{s,d}(i)$

where,

$$\begin{aligned} f - span_{t^{'},t^{'}_2,t^{'}_3}(j,x,y,i) = min\{&span(j,x,t^{'}),\\ &span(x,y,t^{'} + t^{'}_2),\\ &span(y,i,t^{'} + t^{'}_2 + t^{'}_3)\} \end{aligned} \tag{18}$$

So, data path option 1 and ack path option 2 are expected to survive for the entire session, which the ack path option 1 can not. Hence, ack path option 1 will suffer from link breakage. In order to repair the broken link, RREQ packets have to be injected into the network. Let, $\Psi(i,t)$ be the most recent approximate average of downlink cardinality of n_i and it's 1 – hop uplink as well as downlink neighbours till time t.Please note that each node knows 1-hop downlink cardinality of it's 1-hop downlink neighbours from their Ack message and 1–hop uplink neighbours from HELLO messages.

$$\begin{aligned} \Psi(i,t) = [|D_i(t)| + \{ &\sum_{n_j \epsilon D_i(t)} |D_j(t)|\}/(|D_i(t)| + 1)\\ &+\{ \sum_{n_k \epsilon U_i(t)} |D_k(t)|\}/(|U_i(t)| + 1)]/3 \end{aligned} \tag{19}$$

Therefore, cost of flooding by n_i based on latest downlink cardinality information $\Psi(i,t)$ is denoted by C(i,t) and defined by,

$$C(i,t) = \Psi(i,t) + \Psi^2(i,t) + \Psi^3(i,t) + ... + \Psi^H(i,t) \tag{20}$$

where H is the maximum allowable hop count in the network.
So,

$$C(i,t) = \frac{\Psi^{H+1}(i,t) - 1}{\Psi(i,t) - 1} - 1 \tag{21}$$

For maximum energy efficiency of ack option 1, we assume that after flooding, the highest energy efficient path, is discovered. Maximum energy path to the best of n_i's knowledge, consists of a single link $n_\alpha \rightarrow \mathrm{n}_\beta$ s.t. the following two conditions hold :-
i)

$$\frac{P_{max}(\alpha)}{R_\alpha}|_{n_i} \leq \forall_{n_\delta \epsilon \{(n_i) \cup D_i(t_{flood}) \cup U_i(t_{flood})\}} \frac{P_{max}(\delta)}{R_\delta} \tag{22}$$

ii)

$$dist_{\alpha\beta}(t_{flood}) = L|_{n_i} \tag{23}$$

$|_{n_i}$ indicates the relation " as seen by n_i".

where t_{flood} is the timestamp at which RREP arrived at n_α . This RREP corresponds to the RREQs flooded in the network to repair the broken ack option 1. L $|_{n_i}$ is the minimum possible distance between any two nodes, to the best of n_i's knowledge. It is formulated in (22).

$$L|_{n_i} = \forall_{n_x \epsilon \{(n_i) \cup D_i(t_{flood}) \cup U_i(t_{flood})\}} min(dist_{xy}(t_{flood})) \tag{24}$$

and $n_y \epsilon \mathrm{D}_x(t_{flood})$ s.t. $dist_{xy}(t_{flood})$ can be computed by n_i.

Let, ack option 1 broke after acknowledging pac number of packets.
RREP generated by n_d reached n_s at tm_{RREP} and it touched n_j at timestamp $tm_{RCH}(j)$. Information about tm_{RREP} is attached to the first data packet which travels along the data path that consumes least energy. It can not consider energy consumption, in Ack's.

$$del_{d,s}(j) = tm_{RREP} - tm_{RCH}(j) \tag{25a}$$

$$del_{d,s}(x) = tm_{RREP} - tm_{RCH}(x) \tag{25b}$$

$$del_{d,s}(y) = tm_{RREP} - tm_{RCH}(y) \tag{25c}$$

However, ack 1 option 1 is associated to only $del_{d,s}(j)$ not $del_{d,s}(x)$ and $del_{d,s}(y)$.

According to in-tr be time difference between two consecutive ack packets in ack option 1,

$$Pac = \{span(j,i,t^{'}) - del_{d,s}(j)\}/in - tr + 1 \tag{26}$$

Hence, estimated energy consumption $e - ack_1^{'}(j,i,t^{'})$ by ack option 1,corresponding to data path option 1, is formulated in (25).

$$\begin{aligned} e - ack_1^{'}(j,i,t^{'}) &= f_{t^{'}}(j,i) \times pac + c(j,ts) \\ &+ \frac{P_{max}(\alpha)}{R_\alpha}|_{n_i} \times L|_{n_i} \times (pkts - pac) \end{aligned} \tag{27}$$

Here,

$$f_{t^{'}}(j,i) = \frac{P_{max}(j)}{R_j} dist_{ji}(t^{'}) \tag{28}$$

$$ts = t^{'} + del_{d,s}(j) + (pac - 1) \times in - tr \tag{29}$$

For simplicity of computation, we assume that once a path is repaired, it does not break till the session is over.

As far as ack path option 2 is concerned, it is expected to survive till end of the communication session.

$$\begin{aligned} e - ack_2^{'}(j,i,t^{'}) &= \{f_{t^{'}}(j,x) + f_{t^{'}+t_2^{'}}(x,y) \\ &+ f_{t^{'}+t_2^{'}+t_3^{'}}(y,i)\} \times pkts \end{aligned} \tag{30}$$

Based on the values of $e - ack_1^{'}(j,i,t^{'})$ and $e - ack_2^{'}(j,i,t^{'})$, most eligible acknowledgement or MEA for data path option 1 is calculated.

If $e - ack_1^{'}(j,i,t^{'}) \leq$ e-ack$_1^{'}(j,i,t^{'})$ then MEA = 1, else MEA = 2.

Energy consumed in data path option 1 is denoted by $e - dat_1(i,j,t)$ and defined in (29).

$$e - dat_1(i,j,t) = f_t(i,j) \times pkts \tag{31}$$

Overall energy consumed $ov - dat_1(i,j,t,t^{'})$ in data path option 1 corresponds to energy consumption in both data path and it's MEA. Please note that data path started it's operation at time t and MEA at time $t^{'}$.

$$\begin{aligned} ov - dat_1(i,j,t,t^{'}) &= e - dat_1(i,j,t) \\ &+ e - ack_{MEA}^{'}(j,i,t^{'}) \end{aligned} \tag{32}$$

Data path option 2

The scenario of forwarding data and acknowledgement packets in order to complete communication along data path option 2, is shown below :-

data path : $n_i \xrightarrow{t_1} \mathrm{n}_x \xrightarrow{t_2} \mathrm{n}_v \xrightarrow{t_3} \mathrm{n}_j$ (started at time t)

corresponding acks:
1. $n_j \xrightarrow{t_1^{'}} \mathrm{n}_i \xrightarrow{t_2^{'}} \mathrm{n}_x \xrightarrow{t_3^{'}} \mathrm{n}_v$
2. $n_j \xrightarrow{t_4^{'}} \mathrm{n}_x \xrightarrow{t_5^{'}} \mathrm{n}_y \xrightarrow{t_6^{'}} \mathrm{n}_i$ and $n_x \downarrow_{n_v} \mathrm{t}_3^{'}$

Please note that although first Ack corresponding to all data path options is $n_j \rightarrow \mathrm{n}_i$, but n_x and n_v also need to be informed. Therefore, actual ack option 1 is $n_j \rightarrow \mathrm{n}_i \rightarrow \mathrm{n}_x \rightarrow \mathrm{n}_v$.

Similarly, actual ack option 2 is
$n_j \rightarrow$n$_x$ $\rightarrow$n$_y$ $\rightarrow$n$_i$ and $n_x$$\downarrow$n$_v$

As mentioned in case of data path option 1, timestamps written above each $\rightarrow$ sign, specifies the time duration required for communication along the said line. For example, consider the link $n_x \xrightarrow{t_5^{'}}n_y$. Here, $t_5^{'}$ specifies the time span n_x takes to deliver Ack sent by n_j, to n_y. Since n_j transmitted Ack at time $t^{'}$, n_x received it at time ($t^{'} + t_4^{'}$) whereas n_y received the same at time ($t^{'} + t_4^{'} + t_5^{'}$). As mentioned in data path option 1, time required for completion of communication session is $cm_{s,d}(i)$.

Without any loss of generality, we assume that
i) $f - span_{t,t_1,t_2}$(i,x,v,y) $\geq$cm$_{s,d}(i)$
ii)$f - span_{t^{'},t_1^{'},t_2^{'}}$(j,i,x,v) $\leq$cm$_{s,d}(i)$
iii)min$\{f - span_{t^{'},t_4^{'},t_5^{'}}$(j,x,y,i), span(x, v, t$^{'} + t_4^{'})\}$ $\leq$cm$_{s,d}(i)$

Here the situation is that only data path is expected to survive till end of the communication session, and none of the ack options will be able to do so. In that case, assume pac1 and pac2 number of packets will be acknowledged by ack options 1 and 2, respectively, before they break. Mathematical expressions of pac1 and pac2 appear below:-

$$pac1 = \{f - span_{t^{'},t_1^{'},t_2^{'}}(j,i,x,v) - del_{d,s}(j)\}/intr1 + 1 \tag{33}$$

$$pac2 = \{min(f - span_{t^{'},t_4^{'},t_5^{'}}(j,x,y,i), span(x,v,t^{'} + t_4^{'})) - del_{d,s}(j)\}/intr2 + 1 \tag{34}$$

intr1 and intr2 are intervals between acknowledgements of two consecutive packets in ack options 1 and 2, respectively.

$$k = \begin{cases} \text{j if the link } n_j \rightarrow \text{n}_x \text{ breakes in ack option 2} \\ \text{x if the link } n_x \rightarrow \text{n}_y \text{ breakes in ack option 2} \\ \text{y if the link } n_y \rightarrow \text{n}_i \text{ breakes in ack option 2} \end{cases}$$

Estimated energy consumption $e - dat_2(i,j,t)$ of data path option 2 is modelled in (33).

$$e - dat_2(i,j,t) = f_t(i,x) + f_{t+t_1}(x,v) + f_{t+t_1+t_2}(v,j) \tag{35}$$

f has been defined earlier in this subsection.
Similarly, energy consumed in ack paths 1 and 2 are given by $e - ack_2^1(j,i,t^{'})$ and $e - ack_2^2(j,i,t^{'})$ respectively and mathematically expressed in (34) and (35).

$$e - ack_1^2(j,i,t^{'}) = \{f_{t^{'}}(j,i) + f_{t^{'}+t_1^{'}}(i,x) + f_{t^{'}+t_1^{'}+t_2^{'}}(x,v)\} \times pac1 + c(j,t^{'}) + \frac{P_{max}(\alpha)}{R_\alpha}|_{n_j} \times L|_{n_j} \times (pkts - pac1) \tag{36}$$

$$e - ack_2^2(j,i,t^{'}) = \{f_{t^{'}}(j,x) + f_{t^{'}+t_4^{'}}(x,y) + f_{t^{'}+t_4^{'}+t_5^{'}}(y,i) + f_{t^{'}+t_4^{'}}(x,v)\} \times pac2 + c(k,tms) + \frac{P_{max}(\alpha)}{R_\alpha}|_{n_k} \times L|_{n_k} \times (pkts - pac2) \tag{37}$$

$$tms = \begin{cases} \text{A, if the link } n_j \rightarrow \text{n}_x \text{ in ack option 2 breaks} \\ \text{B, if the link } n_x \rightarrow \text{n}_y \text{ in ack option 2 breaks} \\ \text{C, if the link } n_y \rightarrow \text{n}_i \text{ in ack option 2 breaks} \end{cases}$$

where,
A = $t^{'}$ + $del_{d,s}(j)$ + $(pac2 - 1)intr2$
B = $t^{'}$ + $t_4^{'}$ + $del_{d,s}(x)$ + $(pac2 - 1)intr2$
C = $t^{'}$ + $t_4^{'}$ + $t_5^{'}$ + $del_{d,s}(y)$ + $(pac2 - 1)intr2$

Significance of α and L has been described earlier.
Please note that, line breakage in ack option 1 may take place only when link breaks from n_j to n_i. Other links $n_i \rightarrow$n$_x$, $n_x \rightarrow$n$_v$ must be stable till the communication session is over because they are part of the data path expected to survive till the current session completes.

So, only n_j may face line breakage.

Based on this computation, must eligible acknowledgement on MEA for a data path option, is determined as follow :-
If $e - ack_2^2(j,i,t^{'})$ $\leq$e-ack$_1^2(j,i,t^{'})$
then MEA = 1, else MEA = 2.
Overall energy consumption $ov - dat_2(i,j,t,t^{'})$ of data path option 2 is as follows :-

$$ov - dat_2(i,j,t,t^{'}) = e - dat_2(i,j,t) + e - ack_{MEA}^2(j,i,t^{'}) \tag{38}$$

Data path option 1 is selected (i.e. MEP =1) if $ov - dat_1(i,j,t,t^{'})$ $\leq$ov-dat$_2(i,j,t,t^{'})$. Else, data path option 2 is selected(i.e. MEP = 2).

What will happen if a data path breaks ?

Without any loss of generality, assume that
i) $span(i,j,t)$ $\geq$cm$_{s,d}(i)$
ii) $f - span_{t^{'},t_1^{'},t_2^{'}}$(i,x,v,j) $< cm_{s,d}(i)$

Here we see that data path option 1 is expected to survive till end of the communication session which data path option 2 can not. In this case, if both Ack options of data path option 2 survive for a longer time interval than data path 2, still those Ack paths have no significance if the associated data path breaks, irrespective of the fact that no link breakage in ack paths are estimated to occur. On the other hand, if at least one of those Ack paths live shorter than data path 2 and get selected as MEA, then at least two link breakages will occur, first for MEA, then for data path option 2. Case – 1 Both acks survive till the session completes.

Case 1 : Both acks survive till the session completes

Both acks will survive till end of the session
i) $span(j,i,t^{'})$ $\geq$cm$_{s,d}(i)$

ii) $f - span_{t',t'_4,t'_5}$(j,x,y,i) $\geq$cm$_{s,d}(i)$

$e - dat_1(i,j,t)$ is modeled in (29).

$$e - ack_1^{'}(j,i^{'},t^{'}) = f_{t'}(j,i) \times pkts \tag{39}$$

Expression for $e - ack_2^{'}(j,i,t^{'})$ already appears in (28). Selection of MEA is based on the comparison of $e - ack^{'}(j,i,t^{'})$ and $e - ack_2^{'}(j,i,t^{'})$ which even is lesser. $ov - dat_1(i,j,t,t^{'})$ is computed as in (30).

As far as data path option 2 is concerned, assumes that pac3 number of packets are successfully delivered before the path breaks.

$$pac3 = \{f - span_{t,t_1,t_2}(i,x,v,j) - del_{s,d}(i)\}/intr + 1 \tag{40}$$

$$\begin{aligned} e - dat_2(i,j,t) = \{f_t(i,X) + f_{t+t_1}(X,V) + f_{t+t_1+t_2}(V,j)\} \\ \times pac3 + c(\eta, tms1) + \frac{P_{max}(\alpha)}{R_\alpha}|_{n_\eta} \\ \times L|_{n_\eta} \times (pkts - pac3) \end{aligned} \tag{41}$$

$$\eta = \begin{cases} \text{i if the link } n_i \rightarrow \text{n}_X \text{ breakes in data path option 2} \\ \text{x if the link } n_X \rightarrow \text{n}_V \text{ breakes in data path option 2} \\ \text{v if the link } n_V \rightarrow \text{n}_j \text{ breakes in data path option 2} \end{cases}$$

$$tms1 = \begin{cases} \text{t + } del_{s,d}(i) \text{ + (pac3 - 1)intr if } \eta = i \\ \text{t + } t_1 \text{ + } del_{s,d}(i) \text{ + (pac3 - 1)intr if } \eta = X \\ \text{t + } t_1 \text{ + } t_2 \text{ + } del_{s,d}(i) \text{ + (pac3 - 1)intr if } \eta = V \end{cases}$$

$$\begin{aligned} e - ack_1^2(j,i,t^{'}) = \{f_{t'}(j,i) + f_{t'+t'_1}(j,X) + \\ f(X,V)\} \times pac3 + \frac{P_{max}(\alpha)}{R_\alpha}|_{n_j} \\ \times L|_{n_j} \times (pkts - pac3) \end{aligned} \tag{42}$$

$$\begin{aligned} e - ack_2^2(j,i,t^{'}) = \{f_{t'}(j,X) + f_{t'+t'_4}(X,Y) + \\ f_{t'+t'_4+t'_5}(Y,i) + f_{t'+t'_4}(X,V)\} \times pac3 + \\ \frac{P_{max}(\alpha)}{R_\alpha}|_{n_j} \times L|_{n_j} \times (pkts - pac3) \end{aligned} \tag{43}$$

MEA is obtained by comparing $e - ack_1^2(j,i,t^{'})$ and $e - ack_2^2(j,i,t^{'})$. The ack option consuming lesser energy, is elected.
$ov - dat_2(i,j,t,t^{'})$ appears in (36).
Hence we have assumed that after repairing data path, both data and ack options will suffer energy consumption as seen by n_j and paths will be single hop in both direction.

Case 2 : A least one ack path will break after breaking of the fragile data path

As in case 1.

Case 3 : At least one Ack path is expected to break before breaking of the fragile data path

Let, $span(j,i,t^{'})$ $\geq$cm$_{s,d}(i)$
$f - span_{t',t'_4,t'_5}$(j,x,y,i) $<$ $f - span_{t,t_1,t_2}$(i,x,v,j)

Expression for $e - dat_2(i,j,t)$ appears in (39). Assume that ack option 2 will break after acknowledging pac4 number of packets. Please note that $e - ack_1^2(j,i,t^{'})$ is modulated in (40).

$$\begin{aligned} pac4 = min\{f - span_{t',t'_4,t'_5}(j,x,y,i), \\ span(x,v,t^{'} + t_4^{'}) - del_{d,s}(j)\}/intr2 + 1 \end{aligned} \tag{44}$$

$$\begin{aligned} e - ack_2^2(j,i,t^{'}) = \{f_{t'}(j,X) + f_{t'+t'_4}(X,Y) + \\ f_{t'+t'_4+t'_5}(Y,i) + f_{t'+t'_4}(X,V)\} \times pac4 + \\ c(k,tms) + \frac{P_{max}(\alpha)}{R_\alpha}|_{n_k} \times L|_{n_k} \\ \times (pac3 - pac4) + \frac{P_{max}(\alpha)}{R_\alpha}|_{n_k} \\ \times L|_{n_k} \times (pkts - pac3) \end{aligned} \tag{45}$$

i.e.

$$\begin{aligned} e - ack_2^2(j,i,t^{'}) = \{f_{t'}(j,X) + f_{t'+t'_4}(X,Y) + \\ f_{t'+t'_4+t'_5}(Y,i) + f_{t'+t'_4}(X,V)\} \times pac4 + \\ c(k,tms) + \frac{P_{max}(\alpha)}{R_\alpha}|_{n_k} \times L|_{n_k} \times (pkts - pac4) \end{aligned} \tag{46}$$

After comparing $e - ack_1^2(j,i,t^{'})$ and $e - ack_2^2(j,i,t^{'})$, MEA is elected. From this, overall energy consumed by the data path can be computed as prescribed earlier.

4. Illustration With Examples

Please refer to Table 5. All nodes are moving to the left with velocity 1m/s; corresponding connectivity diagram is shown in fig 7.

Example 1

$dist_{iv}(t)$ = $\sqrt{5^2 + 3^2}$ = $\sqrt{25 + 9}$ = 5.83095
$dist_{ij}(t)$ = $\sqrt{3^2 + 3.7^2}$ = $\sqrt{9 + 13.69}$ = 4.763
$dist_{jv}(t)$ = $\sqrt{2^2 + 6.7^2}$ = $\sqrt{4 + 44.89}$ = 6.992

$dist_{iv}(t)$ = 5.83095 = $dist_{vi}(t)$
$dist_{ij}(t)$ = 4.763
$dist_{jv}(t)$ = 6.992
t is the current time, as per direction of movement of the nodes, all links are expected to live for ever with respect to velocity, at least till the present communication session completes.
lifespan (n_i $\rightarrow$n$_v$) = $min(\frac{25}{2}, \frac{20}{2})$ = 10
= lifespan (n_v $\rightarrow$n$_i$)
lifespan (n_i $\rightarrow$n$_j$) = $min(\frac{25}{2}, \frac{35}{2})$ = 12.5
lifespan (n_j $\rightarrow$n$_v$) = $min(\frac{35}{2}, \frac{20}{2})$ = 10

Let the wireless signal travels ϕm in 1 ms. Based on this, per packet delay of links are computed below:-

per packet delay $(n_i \rightarrow n_v)$ = $\frac{5.83095}{\phi}$ms
= per packet delay $(n_v \rightarrow n_i)$
per packet delay $(n_i \rightarrow n_j)$ = $\frac{4.763}{\phi}$ms
per packet delay $(n_j \rightarrow n_v)$ = $\frac{6.992}{\phi}$ms

Assume that, $\phi^{'}$ number of packets are to be transmitted in the session.

Time to live $(n_i \rightarrow n_v)$ = 5.83095($\frac{\phi^{'}}{\phi}$)ms
= Time to live $(n_v \rightarrow n_i)$
Time to live $(n_i \rightarrow n_j)$ = 4.763($\frac{\phi^{'}}{\phi}$)ms
Time to live $(n_j \rightarrow n_v)$ = 6.992($\frac{\phi^{'}}{\phi}$)ms
without any loss of generality, we assume the followings:-
i) 5.83095($\frac{\phi^{'}}{\phi}$) $<$ 10
ii) 4.763($\frac{\phi^{'}}{\phi}$) $<$ 12.5
iii) 6.992($\frac{\phi^{'}}{\phi}$) $<$ 10

i.e. ($\frac{\phi^{'}}{\phi}$) $<$ ($\frac{10}{6.992}$ = 1.43020)

Therefore, all the three relevant links are going to survive till end of the session.
Two data paths exist here from $n_i \rightarrow n_v$. They are $n_i \rightarrow n_v$ and $n_i \rightarrow n_j \rightarrow n_v$. Hence, we do not have any provision to select MEA, because only one choice is available for ack and that is the link $n_v \rightarrow n_i$, costs of data path calculated in examples are per packet cost.

Data Path Option 1

cost$(n_i \rightarrow n_v)$ = 50 × $(\frac{5.83095}{25})^2$ = 50 × 0.0544 = 2.72
Relevant ack is $n_v \rightarrow n_i$.
cost$(n_v \rightarrow n_i)$ = 1 × $(\frac{5.83095}{6})^2$ = 0.944
Overall cost = 3.6644

Data Path Option 2

cost$(n_i \rightarrow n_j \rightarrow n_v)$
= cost$(n_i \rightarrow n_j)$ + cost$(n_j \rightarrow n_v)$
= 50 × $(\frac{4.763}{25})^2$ + 1 × $(\frac{6.992}{8})^2$ = 50 × 0.0363 + 1 × 0.764
= 1.815 + 0.764 = 2.579
Relevant ack = $n_v \rightarrow n_i \rightarrow n_j$
cost = 0.944 + 1.815 = 2.759
overall cost = 5.338

Please note that, this particular example shows that although cost of data path option 1 is higher than cost of data path option 2, still actual overall cost of data path 1 is lesser than overall cost of data path 2. This happened because cost of MEA of data path 2 is significantly higher than cost of MEA of data path 1.

Example 2

$dist_{iv}(t) = \sqrt{5^2+3^2} = \sqrt{25+9}$ = 5.83095
$dist_{jv}(t) = \sqrt{5^2+3^2} = \sqrt{25+25}$ = 7.07106
$dist_{ik}(t) = \sqrt{20^2+3^2} = \sqrt{400+9}$ = 20.2237
$dist_{iv}(t) = \sqrt{5^2+3^2} = \sqrt{25+9}$ = 5.83095
$dist_{ij}(t)$ = 2 and $dist_{kv}(t)$ = 15
t is the current time.

EPT of n_i is shown below in Table 3. For simplicity of computation, here we assume that source transmits one data packet, only after ack of it's predecessor packet has arrived from destination.
The link between n_i and n_j will survive ideally for ever, because both are moving in same direction with same velocity. Similar is the case for link between n_k and n_v. Maximum distance that n_k has to cover to get out of the radio range is (R_i - $dist_{ik}(t)$) where t is the current time. As far as n_v is concerned, maximum distance to be travelled yet to get out of the radio-range of n_i is, (R_i + $dist_{iv}(t)$). Similarly, other links can be explored and characterized.
lifespan $(n_i \rightarrow n_v)$ = $min(12.5, \frac{25+5.83095}{1-0.5})$ = 12.5
lifespan $(n_i \rightarrow n_j \rightarrow n_v)$ = $min(12.5, \infty, \frac{8+7.07106}{1-0.5})$ = 12.5
lifespan $(n_i \rightarrow n_k \rightarrow n_v)$ = $min(10, \frac{25+20.2237}{1+0.5}, \infty)$ = 10
lifespan $(n_v \rightarrow n_i)$ = $min(12.5, \frac{8+5.83095}{0.5})$ = 12.5
lifespan $(n_v \rightarrow n_j \rightarrow n_i)$= $min(12.5, \frac{8+7.07106}{0.5}, \infty)$ = 12.5

As mentioned in example 1, wireless signal travels ϕm in 1 ms.

Per packet delay $(n_i \rightarrow n_v)$ = ($\frac{5.83095}{\phi}$)ms
Per packet delay $(n_i \rightarrow n_j \rightarrow n_v)$ = ($\frac{9.07106}{\phi}$)ms
Per packet delay $(n_i \rightarrow n_k \rightarrow n_v)$ = ($\frac{35.2237}{\phi}$)ms
Per packet delay $(n_v \rightarrow n_i)$ = ($\frac{5.83095}{\phi}$)ms
Per packet delay $(n_v \rightarrow n_j \rightarrow n_i)$ = ($\frac{9.07106}{\phi}$)ms

Let $\phi^{'}$ be the number of packets to be transmitted by the current session.

Time to live $(n_i \rightarrow n_v)$ = 5.83095($\frac{\phi^{'}}{\phi}$)
Time to live $(n_i \rightarrow n_j \rightarrow n_v)$ = 9.07106($\frac{\phi^{'}}{\phi}$)
Time to live $(n_i \rightarrow n_k \rightarrow n_v)$ = 35.2237($\frac{\phi^{'}}{\phi}$)
Time to live $(n_v \rightarrow n_i)$ = 5.83095($\frac{\phi^{'}}{\phi}$)
Time to live $(n_v \rightarrow n_j \rightarrow n_i)$ = 9.07106($\frac{\phi^{'}}{\phi}$)

without any loss of generality, let $\frac{\phi^{'}}{\phi}$ = $\frac{1}{3}$

So, Time to live $(n_i \rightarrow n_v)$ = 1.94365 $<$ 12.5
Time to live $(n_i \rightarrow n_j \rightarrow n_v)$ = 3.02368 $<$ 12.5
Time to live $(n_i \rightarrow n_k \rightarrow n_v)$ = 11.7412 $>$ 10
Time to live $(n_v \rightarrow n_i)$ = 1.94365 $<$ 12.5
Time to live $(n_v \rightarrow n_j \rightarrow n_i)$ = 3.02368 $<$ 12.5

Considering communication from $n_i \rightarrow n_v$,
data path option 1 : $n_i \rightarrow n_v$
ack option 1 : $n_v \rightarrow n_i$
ack option 2 : $n_v \rightarrow n_j \rightarrow n_i$

data path option 2 : $n_i \rightarrow n_j \rightarrow n_v$
ack option 1 : $n_v \rightarrow n_j \rightarrow n_i$
ack option 2 : $n_v \rightarrow n_i \rightarrow n_j$

data path option 3 : $n_i \rightarrow n_k \rightarrow n_v$
ack option 1 : $n_v \rightarrow n_i \rightarrow n_k$
ack option 2 : $n_v \rightarrow n_j \rightarrow n_i \rightarrow n_k$

Time to live ($n_v \rightarrow n_i \rightarrow n_j$)
= Time to live ($n_v \rightarrow n_i$) + Time to live ($n_i \rightarrow n_j$)
= (1.94365 + 2/3)
= 2.6103166 $<$ 12.5

Therefore, the links $n_i \rightarrow n_v$, $n_v \rightarrow n_i$, $n_v \rightarrow n_j \rightarrow n_i$, $n_i \rightarrow n_j \rightarrow n_v$, $n_v \rightarrow n_i \rightarrow n_j$ and $n_v \rightarrow n_j \rightarrow n_i$ are all going to survive fill the session completes.

$n_i \rightarrow n_k \rightarrow n_v$ will break after delivering $\{\phi(\frac{10}{11.7412})\}$ i.e. 0.85ϕ number of packets. In order to deliver the rest i.e. 0.15ϕ number of packets, link repair through injection of RREQ packets need to be done.

Time to live ($n_v \rightarrow n_i \rightarrow n_k$)
= Time to live ($n_v \rightarrow n_i$) + Time to live ($n_i \rightarrow n_k$)
= $(1.94365 + \frac{20.2237}{3})$ms
= (1.94365 + 6.7412)ms
= 8.68488 ms $<$ 10

So, ack option 1 of data path option 3 will break after acknowledging 0.353ϕ packets after which route recovery will be necessary.

Time to live ($n_v \rightarrow n_j \rightarrow n_i \rightarrow n_k$)
= Time to live ($n_v \rightarrow n_j \rightarrow n_i$) + Time to live ($n_i \rightarrow n_k$)
= (3.02368 + 6.7412)ms
= 9.76488 ms $<$ 10 ms

Hence, both ack options 2 of data path option 3 will survive till the data path breaks.

Data Path Option 1

cost($n_i \rightarrow n_v$) = 20 $\times$ $(\frac{5.83095}{25})^2$ = 1.088
cost($n_v \rightarrow n_i$) = $(\frac{5.83095}{8})^2$ = 0.531249
cost($n_v \rightarrow n_j \rightarrow n_i$) = $(\frac{5.83095}{25})^2$ + 0.5 $\times$ $(\frac{2}{8})^2$
= 0.8125
MEA = 1
Hence, overall-cost($n_i \rightarrow n_v$) = 1.088 + 0.531249 = 1.61925

Data Path Option 2

cost($n_i \rightarrow n_j \rightarrow n_v$) = 20 $\times$ $(\frac{2}{25})^2$ + 0.5 $\times$ $(\frac{7.07106}{8})^2$
= 0.5186
cost($n_v \rightarrow n_j \rightarrow n_i$) = 0.8125
cost($n_v \rightarrow n_i \rightarrow n_j$) = $(\frac{5.83095}{8})^2$ + 20 $\times$ $(\frac{2}{25})^2$
= 0.6592
MEA = 2
Hence, overall-cost($n_i \rightarrow n_j \rightarrow n_v$) = 0.5186 + 0.6592 = 1.1778

Data Path Option 3

Here, flooding will be required to repair routes. We simplify computation of cost of flooding by assuming that each node will receive flooded message and broadcast to all 1 – hop downlink neighbours. So, in our example where only four nodes n_i, n_j, n_k and n_v are there, cost of flooding = (3 + 2 + 1 + 2) = 8.

cost($n_i \rightarrow n_k \rightarrow n_v$)
= $\{20 \times (\frac{20.2237}{25})^2 + 10 \times (\frac{15}{16})^2\} \times 0.85 + 8 +$
$(\frac{0.5}{8}) \times 2 \times 0.15$
= 21.8679 $\times$ 0.85 + 8 + 0.01875
= 18.5877 + 8 + 0.01875= 26.6064

cost($n_v \rightarrow n_i \rightarrow n_k$)
= (0.531249 + 13.0879)
= 13.6191

cost($n_v \rightarrow n_j \rightarrow n_i \rightarrow n_k$)
= (0.8125 + 13.0879) = 13.9004

MEA = 1
overall-cost($n_i \rightarrow n_k \rightarrow n_v$) = (26.6064 + 13.619) = 40.2235

Therefore, selected data path is data path option 2 with MEA $n_v \rightarrow n_i \rightarrow n_j$.

Improvement of 2nd data path over 1st data path is (1.61925 - 1.1778)/1.61925 $\times$ 100% i.e. 27.26% and the same of 2nd data path over 3rd data path is (40.2235 - 1.1778)/40.2235 $\times$ 100% i.e. 97.07%.

This example clearly states how important it is to consider lifetime of links w.r.t, one particular communication session. Similar computation can be repeated for communication from n_i to n_j and n_k, too. Modified EPT of n_i after computing MEP for communication from n_i to n_v, is shown in Table 4. ($G^{'}$ of the network will consist of all links appearing in modified EPTs of all nodes in the network)

5. Sleeping Strategy in LNE^2S

Consider fig 9 where we consider uplink and downlink neighbourhood of a node n_i. Current uplink neighbours of n_i are n_p, n_q, n_r and n_s whereas current downlink neighbours of n_i are n_j, n_k and n_l. Ongoing communication sessions are passing through the following links:
i)...... $\rightarrow n_p \rightarrow n_i \rightarrow n_k \rightarrow$ ....
$ii)$...... $\rightarrow n_p \rightarrow n_i \rightarrow n_k \rightarrow$ ....

n_l is a downlink neighbour of n_i at current time t s.t. the following conditions are satisfied :

i) n_l $\epsilon \mathrm{D}_p(t)$
ii) n_j, n_k $\epsilon \mathrm{D}_l(t)$

n_i is allowed to go to sleep provided it is over – exhausted and it's alternative n_l is in a healthy condition, i.e. n_l's estimated residual lifetime is high.
i.e. eng-span(p, l, t) $\geq$ eng-span(p, i, t)
In case of availability of more than one alternative, say n_l and n_m, of n_i, an weight is assigned depending upon relative velocity and residual energy. Weights of n_l and n_m at time t are denoted as W(l,t) and W(m,t) respectively. They are formulated in (45) and (46).
If W(l, t)> W(m, t) then n_l is selected otherwise n_m is selected.

$$W(l,t) = \sum_{n_p \epsilon X_i(t)} (vel - span(p,l,t) - vel - span(p,i,t)) \tag{47}$$

$$W(m,t) = \sum_{n_p \epsilon X_i(t)} (vel - span(p,m,t) - vel - span(p,i,t)) \tag{48}$$

$X_i(t)$ is the number of uplink neighbours of n_i that are engaged in live communication session through n_i. This alternative node strategy of LNE^2S finds suitable replacements of over–exhausted nodes without hampering ongoing communication sessions.

6. Simulation Results

LNE^2S is compared with the energy—efficient techniques SPAN and ANTC using simulation ns-2. Performance metrics are as follows :
i) Network Throughput – It is defined as (Pakt / PAKT)× 100 where pakt is the number of data packets actually delivered to the destination and PAKT is the number of data packets sent by different nodes in the network. Please note that Pakt $\leq$ PAKT.
ii) Average Energy Consumption – This is summation of energy consumed so far by all nodes in the network, i.e. $\sum_{n_i \epsilon N}(Max - eng(i) - res - eng(i)) / \mid N \mid$,where N is the set of all nodes in the network.
iii) Average Per Packet Delay – Let a communication session C started at time $t_C(start)$ and ended at $t_C(end)$. So, average per packet delay of communication in the network is given by $\sum_{C \epsilon S'}(t_C(end) - t_C(start))/(\mid P_C \mid \times \mid S^{'} \mid)$, where $S^{'}$ is the set of all communication sessions in the network and P_C is the set of packets transmitted in the current session C. Simulation Parameters appear in table 5. Simulation results appear in figures 9,10,11,12,13 and 14. Efficiency of two state-of-the art energy efficient techniques SPAN and ANTC are compared with LNE^2S with respect to number of nodes and node velocity. These emphatically express performance enhancement in favour of LNE^2S.

Explanation

With increase in number of nodes, availability of routers grow big. These newly inserted nodes bridge the gap between pairs of network elements situated far apart, giving rise to new routes that did not exist earlier. But once the network is saturated with a huge number of nodes, throughput starts decreasing due to heavy packet congestion and collision. As far as increase in effective average velocity is concerned, it produces fragile links and therefore fragile routes. As a result, lifetime of routes shorten; links tend to break very frequently. In order to repair the broken links, a huge number of RREQ packets are injected into the network. Those packets have to be forwarded by routers consuming additional energy along with increasing packet congestion and collision. More, packets are dropped and lost, reducing network throughput, as seen in figures 9 and 10. As the number of nodes grows big, number of communication sessions also increase generating additional forwarding load. Therefore, average energy consumption monotonically increase. This is evident from fig 11. Unique features of LNE^2S enable it to select those routes for communication that consume minimum energy (along with acknowledgement paths). This saves energy to a great extent. Moreover, the efficient sleeping strategy of nodes allow over-exhausted fellows to go to sleep without hampering ongoing communication sessions. Suitable alternatives are found that will bridge the gap between communicating uplink and downlink neighbours of a node that want to go to sleep. All these features enable LNE^2S to produce much better network throughput, than it's competitors with respect to number of nodes, as evident in fig 9. As far as average velocity is concerned, LNE^2S always give importance to it while selecting routes. Routes with life span higher than lifetime of the communication session, are preferred. As an obvious consequence, number of link breakages reduce even if their average velocity increase and much lesser RREQ packets are injected into the network. This significantly reduces energy consumption and packet loss due to message contention and collision . Hence, average energy consumption is much less in LNE^2S and network throughput is much higher. These claims are graphically supported in figures 10, 11 and 12.
Figures 13 and 14 plot average delay in communication, with respect to number of nodes and average velocity. As already mentioned LNE^2S suffer from much lesser link breakage than it's competitors. Therefore, time to repair broken routes is saved in LNE^2S. This can be seen in figures 13 and 14.

Conclusion

Energy preservation is extremely important for communication in ad hoc networks. LNE^2S preserves energy by selecting optimum route to downlink neighbours of a node elected by the underlying protocol. This prevents network partitioning due to node death and greatly improves network throughput. LNE^2S considers not only energy consumption in data paths but also acknowledgements. This is extremely important because in general, ack paths are different from data paths and links are uni-directional. Even if a link is

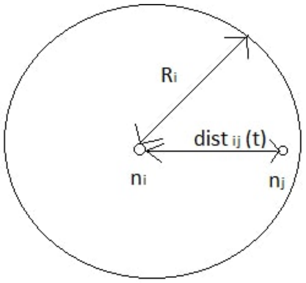

Fig. 1. Transmission power adjustment

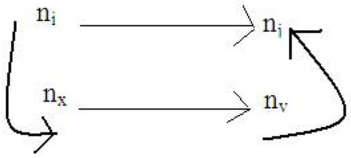

Fig. 2. Traversal of data

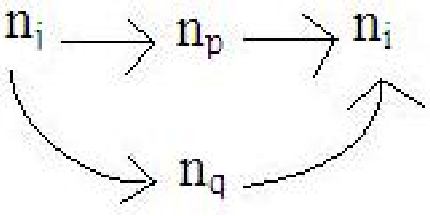

Fig. 3. Traversal of ACK

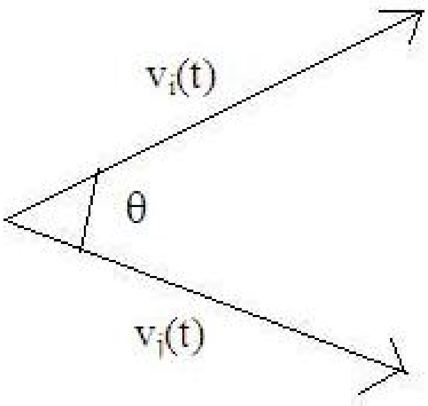

Fig. 4. Computing resultant relative velocity of n_i and n_j

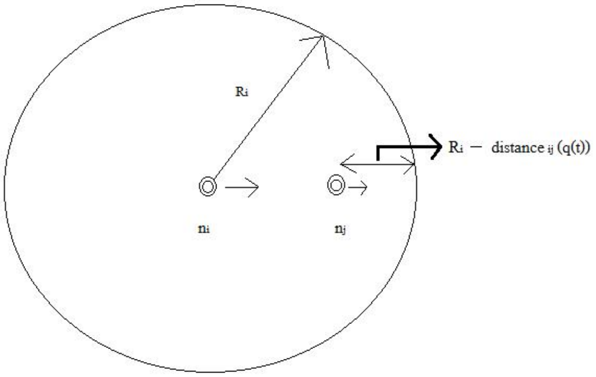

Fig. 5. Link life computation when n_j is moving far from n_j

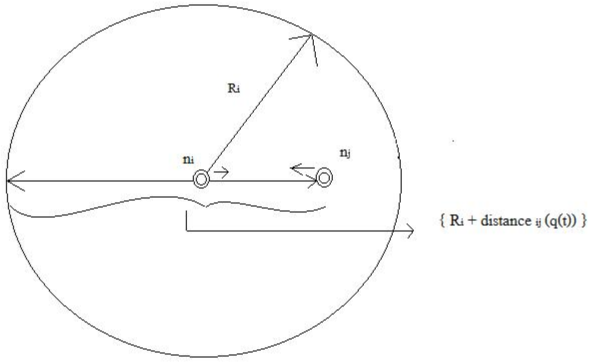

Fig. 6. Link life computation when n_j is coming closer to n_j

bidirectional, energy consumed in a data path and the corresponding ack, may be different due to the difference in maximum transmission power of nodes.

REFERENCES

[1] M. Krunz, A. Muqattash, and S. J. Lee, "Transmission power control in wireless ad hoc networks: challenges, solutions, and open issues," *IEEE Network Magazine*, vol. 18, pp. 8–14, 2004.

[2] M. Chiang, "Balancing transport and physical layers in wireless multihop networks: jointly optimal congestion control and power control," *IEEE Journal on Selected Areas in Communica-tions*, vol. 23, pp. 104–116, 2005.

[3] J. Gomez and A. T. Campbell, "Variable-range transmission power control in wireless ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 87–99, 2007.

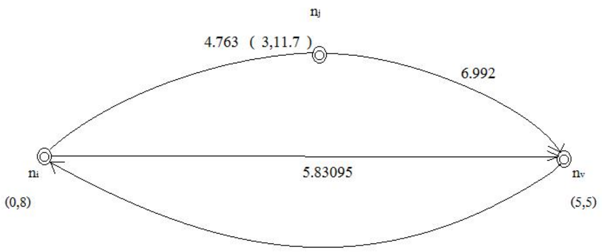

Fig. 7. Connectivity diagram G of a network consisting of n_i, n_j, n_v

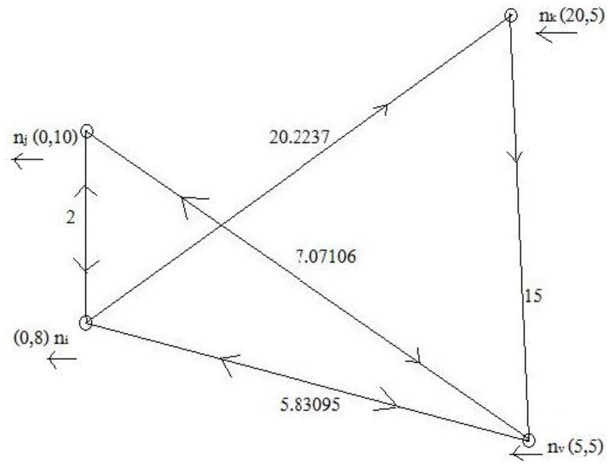

Fig. 8. Connectivity diagram of a network consisting of n_i, n_j, n_k and n_v

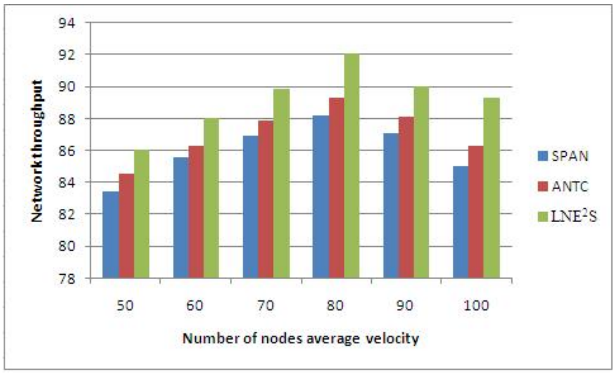

Fig. 9. Network throughput vs number of nodes average velocity = 15 m/s

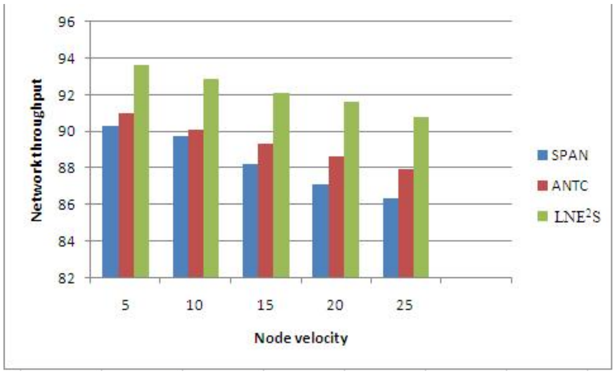

Fig. 10. Network throughput vs node velocity (no. of nodes = 80)

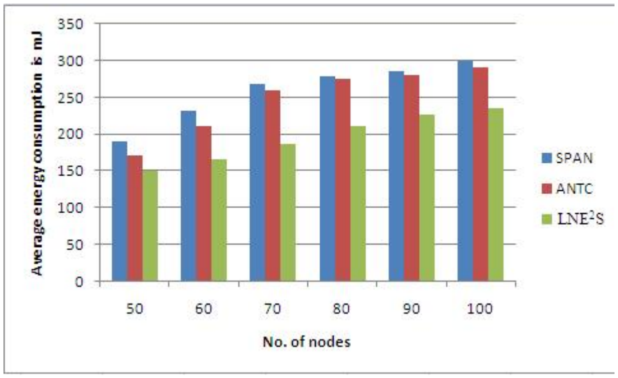

Fig. 11. Average energy consumption is mJ vs no. of nodes (average velocity = 15 m/s)

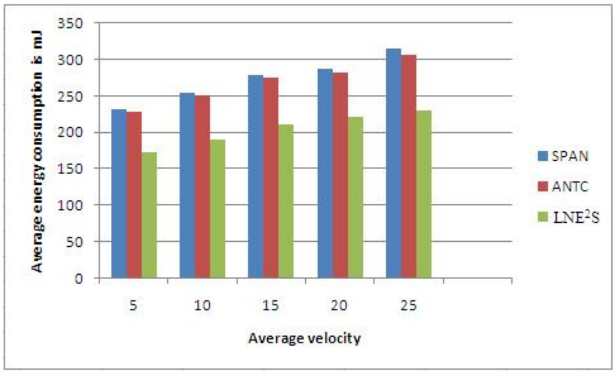

Fig. 12. Average energy consumption in mJ vs average velocity (no. of nodes = 80)

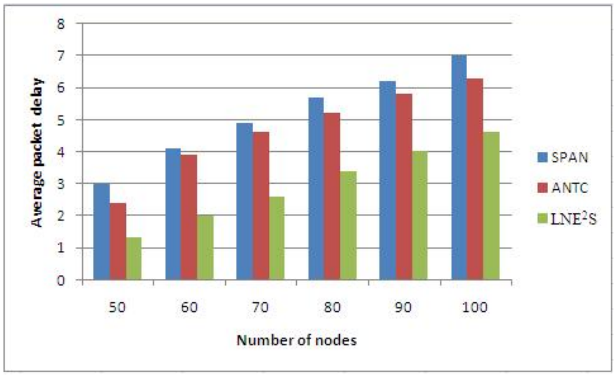

Fig. 13. Average packet delay vs number of nodes(average velocity = 15 m/s)

Downlink neighbour id	Location	Radio range in meter	Max Trans. power in mJ	Max energy in mJ	Rate of energy consumption in mJ/s	Residual energy mJ	MEP,energy cost	MEA,energy cost	Overall cost of MEP
n_i	(0, 8)	25	50	30	2	25	NA	NA, NA	NA, NA
n_j	(3, 11.7)	8	1	45	2	35	NYC, NYC	NYC, NYC	NYC, NYC
n_v	(5, 5)	6	1	30	2	20	NYC, NYC	NYC, NYC	NYC, NYC

TABLE I
EPT OF n_i

(NYC → Not yet computed, NA → Not applicable)

Downlink neighbour id	Location	Radio range in meter	Max Trans. power in mJ	Max energy in mJ	Rate of energy consumption in mJ/s	Residual energy mJ	MEP,energy cost	MEA,energy cost	Overall cost of MEP
n_i	(0, 8)	25	50	30	2	25	NA	NA, NA	NA, NA
n_j	(3, 11.7)	8	1	45	2	35	$n_i \rightarrow n_j$,1.815	$n_j \rightarrow n_v \rightarrow n_i$,1.708	3.523
n_v	(5, 5)	6	1	30	2	20	$n_i \rightarrow n_v$,2.72	$n_v \rightarrow n_i$,0.944	3.6644

TABLE II
MODIFIED EPT OF n_i

Downlink neighbour id	Location	Radio range in meter	Max Trans. power in mJ	Average velocity in m/s	Max energy in mJ	Rate of energy consumption in mJ/s	Residual energy mJ	MEP,energy cost	MEA,energy cost	Overall cost of MEP
n_v	(5, 5)	8	1	$n_v \rightarrow 0.5$	30	2	25	NYC, NYC	NYC, NYC	NYC, NYC
n_k	(20, 5)	16	10	$n_k \rightarrow 0.5$	45	2	20	NYC, NYC	NYC, NYC	NYC, NYC
n_j	(0, 10)	8	0.5	$n_j \rightarrow 1$	30	1	21	NYC, NYC	NYC, NYC	NYC, NYC
n_i	(0, 8)	25	20	$n_i \rightarrow 1$	40	2	30	NA, NA	NA, NA	NA, NA

TABLE III

EPT OF n_i AFTER CONSIDERING COMMUNICATION TO n_v

Downlink neighbour id	Location	Radio range in meter	Max Trans. power in mJ	Average velocity in m/s	Max energy in mJ	Rate of energy consumption in mJ/s	Residual energy mJ	MEP,energy cost	MEA,energy cost	Overall cost of MEP
n_v	(5, 5)	8	1	$n_v \rightarrow 0.5$	30	2	25	$n_i \rightarrow n_j \rightarrow n_v$, 0.518	$n_v \rightarrow n_i \rightarrow n_j$, 0.6592	1.1778
n_k	(20, 5)	16	10	$n_k \rightarrow 0.5$	45	2	20	NYC, NYC	NYC, NYC	NYC, NYC
n_j	(0, 10)	8	0.5	$n_j \rightarrow 1$	30	1	21	NYC, NYC	NYC, NYC	NYC, NYC
n_i	(0, 8)	25	20	$n_i \rightarrow 1$	40	2	30	NA, NA	NA, NA	NA, NA

TABLE IV

MODIFIED EPT OF n_i AFTER CONSIDERING COMMUNICATION TO n_v

TABLE V

COMPUTING $D^{'}_{ij}(t)$ AND RESULTANT VELOCITY IN HORIZONTAL AS WELL AS VERTICAL DIRECTION

	$V_i(t)cos\theta \rightarrow$ R $V_j(t) \rightarrow$ R	$V_i(t)cos\theta \rightarrow$ R $V_j(t) \rightarrow$ L	$V_i(t)cos\theta \rightarrow$ L $V_j(t) \rightarrow$ L	$V_i(t)cos\theta \rightarrow$ L $V_j(t) \rightarrow$ R
n_j is at right side of n_i	$D^{'}_{ij}(t)$= R_i - $dist_{ij}(t)$ RHV = $V_i(t)cos\theta$ - $V_j(t)$ RVV = $V_i(t)sin\theta$	$D^{'}_{ij}(t)$= R_i + $dist_{ij}(t)$ RHV = $V_i(t)cos\theta$ + $V_j(t)$ RVV = $V_i(t)sin\theta$	$D^{'}_{ij}(t)$= R_i + $dist_{ij}(t)$ RHV = $V_i(t)cos\theta$ - $V_j(t)$ RVV = $V_i(t)sin\theta$	$D^{'}_{ij}(t)$= R_i - $dist_{ij}(t)$ RHV = $V_i(t)cos\theta$ + $V_j(t)$ RVV = $V_i(t)sin\theta$
n_j is at left side of n_i	$D^{'}_{ij}(t)$= R_i + $dist_{ij}(t)$ RHV = $V_i(t)cos\theta$ - $V_j(t)$ RVV = $V_i(t)sin\theta$	$D^{'}_{ij}(t)$= R_i + $dist_{ij}(t)$ RHV = $V_i(t)cos\theta$ + $V_j(t)$ RVV = $V_i(t)sin\theta$	$D^{'}_{ij}(t)$= R_i - $dist_{ij}(t)$ RHV = $V_i(t)cos\theta$ - $V_j(t)$ RVV = $V_i(t)sin\theta$	$D^{'}_{ij}(t)$= R_i - $dist_{ij}(t)$ RHV = $V_i(t)cos\theta$ - $V_j(t)$ RVV = $V_i(t)sin\theta$
n_j is directly above or below of n_i	$D^{'}_{ij}(t)$= R_i - $(\frac{R_i-1}{R_i})dist_{ij}(t)$ RHV = $V_i(t)cos\theta$ - $V_j(t)$ RVV = $V_i(t)sin\theta$	$D^{'}_{ij}(t)$= R_i - $(\frac{R_i-1}{R_i})dist_{ij}(t)$ RHV = $V_i(t)cos\theta$ + $V_j(t)$ RVV = $V_i(t)sin\theta$	$D^{'}_{ij}(t)$= R_i - $(\frac{R_i-1}{R_i})dist_{ij}(t)$ RHV = $V_i(t)cos\theta$ - $V_j(t)$ RVV = $V_i(t)sin\theta$	$D^{'}_{ij}(t)$= R_i - $(\frac{R_i-1}{R_i})dist_{ij}(t)$ RHV = $V_i(t)cos\theta$ - $V_j(t)$ RVV = $V_i(t)sin\theta$

TABLE VI

SIMULATION PARAMETERS

Network space	1000m$\times 1000m$
Simulation time	500 second
No. of nodes	50,60,70,80,90,100
Traffic model	CBR
Primary energy of each node	1-10J
Max transmission power	0.5 watt
Min transmission power	0.01 watt
Sleep power	0.01 watt
Packet size	512 bytes
Medium access protocol	IEEE 802.11
Speed of mobile nodes	0-25 m/s

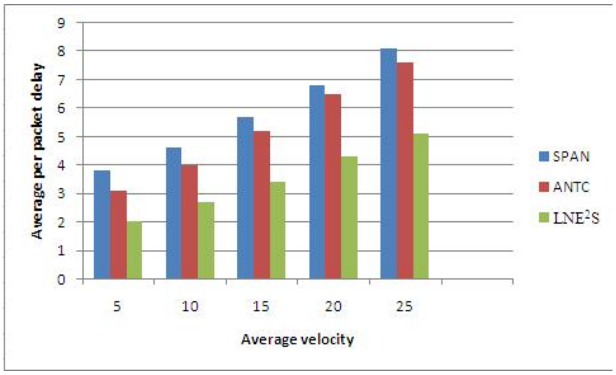

Fig. 14. Average per packet delay vs average velocity (no. of nodes = 80)

[4] X. Zhang, A. Anpalagan, L. Guo, and A. S. Khwaja, "Energy-efficient cooperative mac protocol based on power control in manets," *in Proceedings of the IEEE 29th International Conference on Advanced Information Networking and Applica-tions (AINA 15)*, vol. IEEE, Gwangju, Republic of Korea, pp. 48–53, 2015.

[5] G. De-yun, Z. Lin-juan, and W. Hwang-cheng, "Energy saving with node sleep and power control mechanisms forwireless sensor networks," *The Journal of China Universities of Posts and Telecommunications*, vol. 18, pp. 49–59, 2011.

[6] S. Wu, P. Tseng, and Z. Chou, "Distributed power management protocols for multi-hop mobile ad hoc networks," *Computer Networks*, vol. 47, pp. 63–68, 2005.

[7] R. Zheng and R. Kravets, "On-demand power management for ad hoc networks," *Ad Hoc Networks*, vol. 3, pp. 51–68, 2005.

[8] N. K. Ray and A. K. Turuk, "A review on energy efficient mac protocols for wireless lans," *in Proceedings of the 4th International Conference on Industrial and Information Systems (ICIIS 09)*, pp. 137–141, 2009.

[9] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Computing Surveys*, vol. 37, pp. 164–194, 2005.

[10] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," *in Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 01)*, pp. 70–84, 2001.

[11] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy efficient co-ordination algorithm for topology main-tenance in ad hoc wireless networks," *ACM, Wireless Networks Journal*, vol. 8, pp. 481–494, 2002.

[12] L. Bao and J. J. Garcia-Luna-Aceves, "Stable energy-aware topology management in ad hoc networks," *Ad Hoc Networks*, vol. 8, pp. 313–327, 2010.

[13] P. K. Sahoo, J. Sheu, and K. Hsieh, "Power control based topology construction for the distributed wireless sensor net-works," *Computer Communications*, vol. 30, pp. 2274–2785, 2007.

[14] Z. H. Mir and Y. Ko, "Adaptive neighbor-based topology control pro-tocol for wireless multi-hop networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, 2012.

[15] J. Sheu, S. Tu, and C. Hsu, "Location free topology control pro-tocol in wireless ad hoc networks," *Computer Communications*, vol. 31, pp. 3410–3419, 2008.

[16] J. Gui and A. Liu, "A new distributed topology control algo-rithm based on optimization of delay and energy in wireless networks," *Journal of Parallel and Distributed Computing*, vol. 72, pp. 1032–1044, 2012.

[17] L. Tan and M. Wu, "Data reduction in wireless sensor networks: a hierarchical lms prediction approach," *IEEE Sensors Journal*, vol. 16, pp. 1708–1715, 2016.

[18] L. Tan, X. Zhan, J. Li, and F. Zhao, "A novel tree-based broadcast algorithm for wireless ad hoc networks," *International Journal of Wireless and Mobile Computing*, vol. 1, pp. 156–162, 2006.

[19] S. Zhao, L. Tan, and J. Li, "A distributed energy efficient multicast routing algorithm for wanets," *International Journal of Sensor Networks*, vol. 2, pp. 62–67, 2007.

[20] T. F. Neves and J. L. Bordim, "Topology control in cooperative ad hoc wireless networks," *Electronic Notes in Theoretical Com-puter Science*, vol. 302, pp. 29–51, 2014.

[21] D. Shang, B. Zhang, Z. Yao, , and C. Li, "An energy efficient localized topology control algorithm for wireless multihop networks," *Journal of Communications and Networks*, vol. 16, pp. 371–377, 2014.

[22] K. Moon, D. Yoo, W. Lee, and S. Oh, "Receiver cooperation in topology control for wireless ad-hoc networks," *IEEE Trans-actions on Wireless Communications*, vol. 14, pp. 1858–1870, 2015.

[23] L. M. Feeney, "An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks," *Mobile Networks and Applications*, vol. 6, pp. 239–249, 2001.

[24] P. Siripongwutikorn and B. Thipakorn, "Mobility-aware topol-ogy control in mobile ad hoc networks," *Computer Communi-cations*, vol. 31, pp. 3521–3532, 2008.

[25] R. Ramanthan and R. Hain, "Topology control of multihop wireless networks using transmit power adjustment," *Proc. of IEEE INFOCOM00*, pp. 404–413, 2000.

[26] A. K. Das, R. J. Marks, M. El-Sharkawi, P. Arabshahi, and A. Gray, "Minimum power broadcast trees for wireless networks: Integer programming formulations," *In Proceedings of IEEE INFOCOM*, pp. 2210–2217, 2003.

[27] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Algorithms for energy-efficient multicasting in static ad hoc wireless netoworks," *Mobile Networks and Applications(MONET)6*, pp. 251–263, 2001.