A New Approach for Designing and Analysis of Distributed Routing Algorithm and Protocols of Wireless Ad Hoc Network

Masudul Haider Imtiaz, Mst. Rumana Aktar Sumi,
Dr. S. M. Mostafa Al Mamun
Department of Applied Physics, Electronics &
Communication Engineering
University of Dhaka, Bangladesh.
E-mail: masudul4145@gmail.com

Md. Naeemul Hassan
Department of Computer Science and Engineering
University of Texas at Arlington,
Texas, USA
E-mail: naeemul.hassan@mavs.uta.edu

Abstract— Having no centralized administration and any fixed infrastructure, communication in wireless Ad-Hoc network is more complex than other wired or wireless infrastructure network. Broadcasting based on connected dominating set (CDS) has proved most promising approaches here where only nodes of small dominating set need to relay the broadcast packet, but in the process nodes inside the set consume much more energy than nodes outside to handle various bypass traffic. Existing CDS protocols are not adaptive to such difference in energy consumption by nodes within and outside CDS. This paper demonstrates a non-trivial extension of an efficient CDS-based routing algorithm to create a link with most recent power aware algorithms. Considering energy level of each & every node, it is proposed to apply activity scheduling mechanism. This work also employs such improvement over the different established protocols like DSR, AODV, TORA. Effectiveness of proposed algorithm under different level of nodal mobility is confirmed through simulation.

Keywords- Activity scheduling, Ad hoc Network, D-2 coloring, power_level, rotation, WCDS.

I. INTRODUCTION

An Ad Hoc network is a collection of wireless digital data terminals forming a temporary network without aid of any established networking infrastructure or any centralized administration. Instead, such battery constrained nodes (of finite capacity) need to communicate with each other either directly in bandwidth constraint links or with the help of intermediate nodes i.e. may serve as routers due to the limited range of the wireless transmission. Sensor networks, mobile commerce, disaster recovery, rescue and automated battlefields are examples of application environments. Currently, the emergence of such nomadic application has raised intense research in flexibility and ease in working environment. Energy management strategies can conserve the energy of battery powered subscribers by taking advantage of the energy available at base stations of typical infrastructure network. Development of new power-aware routing algorithm for optimization is critical for practical deployment of these networks. Objective of this paper was to finding out routes that cost minimum power consumption but maintain efficient throughput, lower time & message complexity and secure broadcasting. Having a backbone in any distributed network structure always reduces the setup latency for new connections. This is intuitive since initial route discovery broadcast messages may travel much faster on the backbone than off the backbone and also beneficial for aggregating traffic routing tasks to a few selected nodes. This process reduces the contention at the MAC layer from different neighbors of the same neighborhood that was supposed to increase the power consumption. Therefore, authors were motivated first to create a link between the CDS based and power-aware algorithms considering the attributes of all existing routing protocols.

To prolong the life span of each individual node, hence, the total network without sacrificing system performance this paper intends to modify WCDS (Well Connected Dominating Set), a distributed sub-linear time algorithm introduced by Partha & Gandhi [1] using D2-coloring algorithm. It requires at most O (1) times the number of colors used by an optimal algorithm, O(Δlog2n) running time where Δ is the maximum node degree and n is the size of the network while other existing CDS/MCDS implementations require Ω(n) running time. To minimize the power consumption by backbone nodes, activity scheduling mechanism and rotation of MIS nodes are proposed to add in existing protocol. So, modification has done in almost every stages of original algorithm. Ns-2 simulator was also engaged to prove the effectiveness of such algorithm. Authors further intend to assess this scheme with other traditional algorithms and established protocols to make them power-aware and to experiment on such mobile autonomous networks. Simulation result estimates that this improvement would always possible for different type protocols like DSR, AODV, TORA etc.

II. DESCRIPTION OF WCDS

A set is dominating if every node in the network is either in the set or a neighbor of a node in the set. When a DS is connected through intermediate nodes, it is denoted as a CDS. In a graph, G = (V,E) comprised of a set of vertices V and time-varying edges E, a dominating set is a set of vertices V₀ ∈ V such that every node in V−V₀ is adjacent to some node in V₀ . It may be found considering MIS nodes (maximum independent set that no two vertices within the set are adjacent to each other). Partha & Gandhi denoted WCDS, a special CDS if it satisfies the following structural properties [1]:

- (P1) Low Degree: Let G’=(W,E’) be the graph induced by the nodes in W. For all u ∈ W, let d’(u) denote the degree of u in G’. Then, d’(u) ≤ k₁ (where k₁ is a constant) & ∀u∈ W.

This Paper is an extended version of Conference paper)
(P2) Low Size: Let \(|opt|\) be the size of an MCDS (Minimum Connecting Dominating Set) for an undirected graph, \(G = (V, E)\). Then, \(|W| \leq k_2 \cdot |opt|\); where \(k_2\) constant.

(P3) Low Stretch: Let \(D(p, q)\) denote the length of the shortest path between \(p\) and \(q\) in \(G\). Let \(D_w(p, q)\) denote the length of the shortest path between \(p\) and \(q\) such that all the intermediate nodes in the path belong to \(W\). Here, \(s_w \leq k_3\) where \(k_3\) is a constant if it satisfies the following condition:

\[
s_w = \max \left\{ \frac{D_w(p, q)}{D(p, q)} \right\}
\]

Moreover, Partha & Gandhi proved following applications of this distributed algorithm [1]:

- An online distributed algorithm for collision-free, low latency, low redundancy and high throughput broadcasting.
- Distributed capacity preserving backbone for unicast routing and scheduling.

While D-2 coloring is an assessment of colors to vertices such that every vertex has a color & two vertices which are D-2 neighbor of each other are not assigned the same color [1]. Same color vertices belong to same color class can transmit message simultaneously without any collisions.

### III. RELATED WORK

CDS was first proposed as a prime candidate for virtual backbones in [2,3,4,5]. Guha and Khuller [6] presented centralized approximation algorithms which are guaranteed to produce a CDS of size \(O(\log n)\) times that of an MCDS. Dubashi et al. [7] present a fast distributed CDS algorithm for undirected graphs with a running time of \(O(\log^3 n)\) & size \(O(\log \Delta)\) times MCDS and stretch \(O(\log n)\). Algorithms in [8, 9] construct a CDS of size \(O(n)\) times MCDS and stretch \(O(1)\) in arbitrary undirected graphs with time complexity in [9] is \(O(\Delta^3)\). However, these algorithms are not directly applicable to wireless ad hoc networks as message loss due to collision was not considered. Alzoubi et al. [10] proposed a distributed CDS algorithm for UDGs of size \(O(1)\) times MCDS with \(O(n)\) time and message complexity. Gandhi improved this time complexity by proposing sub-linear time distributed algorithm [1] i.e. constructing a WCDS of size \(O(1)\) times MCDS and \(O(1)\) stretch with drastic decrease in the time complexity (from \(O(n)\) to \(O(\log 2n)\) at the expense of a slight increase in the message complexity from \(O(n)\) to \(O(n \log n)\). Also, utilizing Topkis’s [11] time complexity analysis, this WCDS based online collision-free broadcast algorithm guarantees low latency, low number of re-transmissions and high throughput, all within \(O(1)\) times their optimal values considering the fact that the messages can be lost due to collisions facing in communication process. Besides, several power-aware protocols were proposed in [3], but none of them solved the dependencies imposed on backbone nodes. Ivan & Datta [12] proposed a power & energy aware algorithm and Rong & Kravets [13] planned an on-demand power algorithm but neither of them described the formation of backbone nodes. These power-aware protocols are often based on the following techniques: using variable power (active/standby modes), distributing loads among neighbors, re-transmission avoidance criteria etc. Distribution of loads between neighbor nodes based on traffic and QoS (at network layer) causes bandwidth and message complexity where mode switching between active & standby aims to avoid spending energy during system idle periods & increase in channel utilization & reduction in interference. The problem is complex since the choice of the power level fundamentally affects many aspects of network operation in physical, network, transport layer. According to Feeney and Nilsson in [14], the wireless data interface consumes nearly the same amount of energy in transmit, receive & idle states, whereas in the sleep state, a data interface cannot transmit or receive and thus its power consumption is highly reduced [15]. It was experimentally confirmed by Span [15] that the ratio of energy for transmit, receive, idle, and sleep is 13:9:7:1. However, it is not possible to have a mobile device most of the time in power-saving mode (sleep state), which will extend its battery lifetime but comprise the network lifetime, because ad hoc networks rely on cooperative efforts among participating nodes to deliver messages. A possible strategy could be allowing the network data interface to enter power-saving mode while trying to achieve a minimum impact to the process of sending and receiving messages. Similar technique, activity scheduling was introduced in [16] where it was anticipated to rotate the role among all nodes based on their power levels without any intelligence. Again [17] proved that activity scheduling would be the best power saving policy. Considering all these approaches, authors were motivated to recombine activity scheduling scheme with best backbone formation algorithm to ensure efficient broadcasting of nodes with little energy consumption as well as prolonged network lifetime.

### IV. AUTHOR’S CONTRIBUTION

Authors have succeeded on a non-trivial extension of WCDS algorithm named activity scheduling and rotation of MIS nodes for local solutions and to present their applications in the power saving & as well as broadcast process. These power-aware modifications have done to utilize the characteristics of original WCDS algorithm described by Partha & Gandhi based on D2 coloring [1]. Simulation results from ns-2 simulator supports these ideas. Such backbone approach minimizes overall energy consumption by putting the highest number of nodes in a sleep state. Authors propose here to rotate the role of dominating (active) and non-dominating (sleep) nodes based on energy level of each and every node. Acting new nodes need to perform the scheduled jobs assigned for previous selected one. Every section of proposed algorithm depends on data collected from the physical and MAC layers & proves much more adaptive to topology changes. Authors prove its effectiveness by applying it in established routing protocols.
V. MODIFICATION OF WCDS FORMATION

WCDS formation described by Gandhi & Partha consists of three stages. The first stage involves D2-coloring of the nodes in the network using a list of c colors. The second stage involves constructing of a Maximal Independent Set (MIS) and the third stage involves connecting the MIS. The second and third stages can be easily implemented since transmissions can be scheduled using the D2-coloring computed in the first stage. This algorithm is modified such that: initially all the nodes are assumed to be in full charged state and in failure of providing some mandatory network requirements for power crisis, activity scheduling joins the algorithm to minimize the power consumption at least by the backbone nodes. Hence, the network performance can be improved dramatically. Fourth & fifth stages are added to this algorithm and first three stages are modified accordingly. Activity scheduling will be switched based on two newly defined variables power_level & priority_level.

Stage 1: D2-coloring

From a list of colors, L(u), the coloring algorithm proceeds in a synchronous round by round fashion. Each round consists of four phases: TRIAL, TRIAL-REPORT, SUCCESS and SUCCESS-REPORT. Description is omitted here as Partha & Gandhi [1] gave the details analysis of stage-1. In all phases of stage 1, every node keeps a variable named "power_level" which reflects the remaining charge state of the node. Initially it is assumed all the nodes remain in their full charged state before any kind of communication activities. Every node marks this state as "power_level = 10". After losing some charge, nodes mark this state by lowering power_level proportionally. So, "power_level = 0" indicates the node is now out of charge. Simulation results confirms that almost all nodes do maintain their full charged state in all phases of stage 1 as no big deal of real communication initially.

Stage 2: Constructing the MIS

Stage 1 ensures that all nodes in the network have a valid D2-coloring using c colors {1,2,3,...,c}. During this stage, a Maximal Independent Set (MIS) is built iteratively in c time slots. During slot i, all nodes belonging to color class i attempt to join the MIS. Partha & Gandhi assures that a node joins the MIS if and only if none of its 2-hop neighbors are currently part of the MIS. According to their algorithm, initially nodes compete each other to join MIS. Once a member node of class i win the race, join MIS, eventually other members of same class. After joining the MIS, a node broadcasts a message to its neighbors indicating that it joined the MIS. Nodes transmitting during the same time slot belong to the same color class and hence do not share a common neighbor. Clearly, this stage requires exactly c time steps. This paper proposes to choose MIS intelligently to include less number of nodes in backbone. Color class i which has minimum number of nodes initiates to form MIS. For this purpose, every node maintains a variable "member.color". In stage 1, it was mentioned that, every color is represented by an id. So, every node in same color class can also be represented by a number id in an incremental fashion. For example "member.color=6.5" represents 6th node of color class id 5 (say red color class). So, total member nodes of any color class can easily calculate. First member of a color class having less member nodes initiates MIS forming and then next members of the same color class. For example, node having "member.color=1.5" initiates if this color class id 5 have minimum number of nodes. Moreover, a variable "priority_level" is being added to all nodes forming MIS. Each node corresponding to MIS (color class i) marks its priority_level based on power_level ("priority_level=10" is assumed initially). Actually, for MIS nodes, priority_level = power_level. Other nodes excluding MIS marks "priority_level=0" indicating they are not in the MIS. Fig. 1 shows a D2-coloring algorithm representation where nodes are arranged in such an order that the same color nodes forms the same color class. For simplicity, authors intelligently mark them as that the same color class nodes have the same ones value (e.g. white nodes are marked as 03, 13, 23... while orange nodes are marked as 05, 15, 25.. and vice versa) instead of "member.color notation".

Stage 3: Connecting the MIS

Gandhi & Partha divided this stage into six phases. Each phase is one frame long (length c). As in stage two, nodes transmit only during the time slot corresponding to their D2-color. During the first phase, all MIS nodes transmit a PHASE-1 message consists of just node's ID. In second phase, any node u which received a PHASE-1 message, transmits a PHASE-2 message: a concatenation of ID(u), power_level and all the PHASE-1 messages received by u. In the third phase, any node v which received a PHASE-2 message transmits a PHASE-3 message: a concatenation of ID(u), power_level and all the PHASE-2 messages received by u. By the end of the third phase, every MIS node v knows every other MIS node u in its D3-neighbor-hood. Node v also knows all paths of length at most 3 between itself and u. Node v constructs a PHASE-4 message as follows: for every other MIS node u such that u is in its D3-neighborhood and ID(v) > ID(u). Node v chooses a shortest path of length at most 3 hops between itself and u. It adds its power_level information intelligently to its PHASE-4 message. Initially power_level’s of intermediate

![Figure 1: Representation of D2 coloring of nodes in a sample area](image)
nodes are not taken into account as it is mentioned all the nodes are assumed to have power\_level=10 initially. All MIS nodes transmit a PHASE-4 message during the fourth phase [1]. Every node $u$ which received a PHASE-4 message transmits a PHASE-5 message: a concatenation of all the PHASE-4 messages received by $u$. Finally every node $u$ received a PHASE-5 message transmits a PHASE-6 message: a concatenation of all the PHASE-5 messages received. By the end of this stage, any MIS node $u$ knows the path between itself and any other MIS node $v$ which in its D3-neighborhood.

Stage 4: Activity Scheduling

Authors propose to rotate the role of dominating (active) and non-dominating (sleep) nodes based on power\_level in backbone construction. The localized scheduling/rotation can be as follows: Activity scheduling can be of two types: based on the power\_level of intermediate nodes or based on the priority\_level of MIS nodes. When the intermediate nodes between two MIS nodes found itself in less than the half charged state (i.e. power\_level<5), then it lowers its priority (called tired). Its new priority is propagated to 2-hop neighbors and adjacent MIS nodes get updated with that. MIS node (lower id $u$ than $v$) re-initiates Stages-3, assigning new intermediate nodes i.e. reconnect former MIS node in another path only exchanging three PHASE messages PHASE-4,5,6 messages. Hence, choice of new intermediate nodes is based on two criteria successively:

1. Shortest path between two MIS nodes.
2. Efficient power\_level of intermediate nodes (i.e. power\_level>>5).

Stage 5: Rotation of MIS nodes

If the backbone fails due to lack of power of MIS nodes to respond then the whole networks breaks down. Authors introduce a new scheme: rotation of MIS nodes. When power\_level of a MIS node decreases, its priority\_level also decreases proportionally. In Power\_level <5 situation, it lowers its priority also (called tired). Its new priority is propagated to its 2-hop neighbors and adjacent MIS nodes get updated with that. MIS node (lower id $u$ than $v$) re-initiates Stages-3, assigning new intermediate nodes i.e. reconnect former MIS node in another path only exchanging three PHASE messages PHASE-4,5,6 messages. Hence, choice of new intermediate nodes is based on two criteria successively:

1. Shortest path between two MIS nodes.
2. Efficient power\_level of intermediate nodes (i.e. power\_level>>5).

Stage 5: Rotation of MIS nodes

If the backbone fails due to lack of power of MIS nodes to respond then the whole networks breaks down. Authors introduce a new scheme: rotation of MIS nodes. When power\_level of a MIS node decreases, its priority\_level also decreases proportionally. In Power\_level <5 situation, it lowers its priority also (called tired). Its new priority is propagated to its 2-hop neighbors, next MIS nodes and accordingly the whole network. A better approach would be rotating the roles of dominating (active) and non-dominating (sleep) nodes. Any node in MIS forming class $i$ loses priority, a new class comes into the act of MIS and stages 2 & 3 re-schedules. Nodes of $i$-th class switch themselves in sleep mode to save power. New MIS nodes set priority\_level = power\_level. Other nodes excluding new MIS marks "priority\_level=0" (as usual) indicating that they are not in the new MIS. The rotation of the roles of MIS nodes can be done between two classes intelligently. New color class is chosen depending on the minimum number of member nodes (similar approach of Stage 2). It is assumed that, new color class (say $i_{th}$-th class) can perform all the duties that were scheduled for

Figure 2: Schematic representation of modified WCDS algorithm. Initially, black color class is assumed to have least member nodes. So, black nodes form MIS. Stage-5: (a,b,c,d) connecting MIS nodes by exchanging phase1-6 messages initiates from lower id 11(black) and (e):11-14-17-21 nodes form backbone. Stage-4: (f) power crisis in intermediate node 14, phase-4,5,6 message initiates from node higher black id 21. (g) new backbone forms by 11-13-17-21. Stage-5: (h) power crisis in MIS node 11, in a situation of backbone 11-15-18-21(i,j,k) new color class brown is assigned as it has second minimum member nodes; node10 initiates another phase1-6 message exchange (l) new backbone 10-14-16-20. In all the process of new backbone assignment, previous backbone is still used to transmit messages. When whole backbone formation completes, then switching to new backbone.
class i and in the meantime, the node tired its priority is supposed to recover its power. In future, when the new MIS nodes fail to perform due to lack of power in power_level<5 situation, it propagates the MIS role to new i-th class. And after performing its duty as MIS unless power down, then rotating backed to class i (if it has recovered well). If class i is not ready yet, then rotate the role to i+1-th class or i-1-th and returning back to i-th class & vice versa depending on the power_level & number of member nodes.

VI. ANALYSIS

Any changes of nodes’ status are propagated to only 2-hop neighbors. Here an asynchronous wake up scheme [17, 19] is assumed for communication among neighbors and the propagation delay of each hop is bounded by the scheduling frame T. A WCDS derived can be maintained in a localized way where only nodes in a small vicinity of tired/sleep nodes need to modify their markers.

Rotation Rule:

All active nodes newly marked as TIRED in a rotation process must stay active for additional three scheduling frames (3T) before switching to the sleep mode & extends to 6T in case of MIS rotation.

**Theorem 1:**

In the rotation process, when an intermediate node of WCDS rotates status, no more than 4 nodes within 2 hops of a tired/active/sleep node need to change their status.

**Proof:**

Let u be a tired/active/sleep node presents in WCDS. First to consider parent MIS nodes exploit it before to maintain communication between them. Based on definitions of the D2 coloring & WCDS formation, two MIS nodes (say v1,v2) selected that node previously depending only on the list of u’s 1-hop neighbors, their priorities, and wireless links among them. Both MIS nodes have kept the copy of stage-3 messages and do know all the paths between themselves in D3-neighborhood. Higher id node (say v3) initiates phase-4 message of stage-3 again, re-establish a shortest path based on first three phase messages. This time it avoids that path where power_level of intermediate nodes are under the critical value. Here only three rounds of phase messages necessary to reassign the intermediate nodes. As it is assumed here that a wireless link does not break unless an end node switches off. The assignment of new intermediate nodes of v1 & v2 depends only on the list of v3’s 2-hop neighbors and their priority_levels. After excluding the impact of wireless links based on previous assumption, it is proved that at most two nodes may be excluded and two nodes may included in the backbone, so, highest four nodes here need to change their status based on power_level. This theorem shows that a tired/on/off node affects only highest 3 nodes around and the process converges only after three rounds of “phase message” exchanges, which means a handover interval of 3T. For a smooth handover, the following rule is used to preserve a WCDS during the rotation process.

**Theorem 2:**

The rotation rule preserves a WCDS during the rotation and also after completing rotation process.

**Proof:**

Let C(t) be the set of active nodes at time t. Assume the rotation process starts at t0. By definition, C(t0), the set of marked nodes is an WCDS for t ≤ t0. By Theorem 1, the rotation process converges no later than t0 + 3T i.e. C(t) is an WCDS for t ≥ t0 + 3T & C(t(t0) ∈ C(t) for t ∈ [t0, t0 +3T] by rotation rule. Because C(t) is assumed an WCDS, C(t0) would maintain WCDS characteristics during this period. After rotating intermediate nodes, it also remains in WCDS (proof omitted). Similarly, in the process of shuffling MIS nodes to a new color class, rotation rule preserves WCDS during all 6T time frames. When new MIS class and all of its intermediate nodes are confirmed, only then jobs would be exchanged accordingly between them.

VII. EVALUATION BY SIMULATION

This anticipated algorithm is implemented manually in ns2 simulator. In this process, all the nodes in an schematic environment of 100mx100m are marked with appropriate color class (D2 coloring of stage 1 is being applied) & nodes arrange backbone by themselves. Applying Bash script that at every 200th sec, a power crisis scenario introduces in one of the intermediate nodes ( i.e. nodes excluding MIS). Intermediate nodes have changed their marks and new backbone has formed accordingly. After another 200sec, another power crisis arises & intermediate nodes are changed again. At about 800th sec, MIS nodes rotate & new color class comes into act as one of MIS nodes failed to respond due to a forced power crisis situation. This process continues for a total of 1200 sec (20 min). From simulation result, a GNU plot of time vs. average throughput is drawn. Dramatic results have observed that average throughput (calculated in 10sec interval) decreases for only a small period of time (around 20 sec) when intermediate nodes are changed in every 200th sec. As soon as, new nodes are assigned, throughput increases accordingly.

When MIS nodes rotate, throughput decreases in larger amount as expected but presents a much improved result than other algorithms. In a dense environment, better solution is found for same criteria i.e. throughput is much more stable and only a small amount of fluctuation is observed in rotation periods. Fig. 3 shows a comparison of scenarios having different number of nodes. In denser environment, Average throughput is well over 150 Mbps which is a great...
improvement in any case: rural, sub-urban or urban. These plots ensure that communication between such battery constrained nodes can extend for maximum period of time until battery-power of all nodes have dried out completely. Hence, battery-dependencies are handled more efficiently than other existing power-aware algorithms without sacrifice performance like average throughput or others.

VIII. USING THE TEMPLATE IMPLEMENTATION IN ESTABLISHED PROTOCOLS

Typical broadcasting protocols are normally not adaptive to topology or route changes or no solution when large geographical range, not adaptive to buffer overflow problem. Author tried to implement the previous idea to different types of routing protocols. Proactive type algorithm (table driven routing protocols) like DSDV and reactive type AODV as on demand routing protocol and TORA are modified here to prove the effectiveness of the proposed algorithm.

A. DSDV (Destination Sequence Distance Vector)

DSDV is a distance vector routing protocol [20] where each nodes need to maintain a routing table to indicate each destination i.e. next hop & number of hops to reach that destination. Authors propose to add the power_level parameter in that table. Each node periodically broadcasts routing updates associated with their power_level. The sequence number being used previously to tag each route is modified to provide with respect to both shortest path and the power_level of intermediate nodes. Hence, the sequence number will show the freshness of the route and the energy level of intermediate nodes. A route with highest sequence number is more favorable as the higher sequence number nodes will indicate the route with maximum charged nodes and vice versa. Among two routes with same sequence number, the one with less hop will more favorable like previous. Here, activity scheduling may also be imposed. When any node find itself in a power_level<5 criteria, it retires temporarily from the communication, alarms its neighbor nodes. So, associated nodes detect that the route is broken, its hop number will be set to infinity and sequence number will be assigned an odd number: as even number will correspond to sequence number of connected path as described in original DSDV protocol. Simulation result shows that this approach minimizes the basic bandwidth limitation and stale routing of original DSDV protocol. Earlier this protocol was vastly used for small network of high mobility and large network of low mobility. These modifications can make this protocol suitable for medium type network with high mobility and handling of control messages become easier. Here, less delay is involved in the route set up process and easier to implement.

B. AODV (Ad-hoc On Demand Distance Vector)

AODV originally does not require maintaining routes to destination that are not actively used rather employ RREQ (route request), RREP (route reply) and RERR (route error) messages via UDP (update packet). Like DSDV, destination sequence number for each active route is being maintained for any route request from neighbor nodes that provides route freshness and loop freedom. Author modified this on-demand protocol as that, when a node wants to find a route to another one, it broadcasts a RREQ with power_level status to the entire network till either a destination is reached or another node is found with a “fresh enough” route to that destination. Then it checks all the power_level parameter of intermediate nodes from active neighbor lists. Then, a RREP is sent back to the source and the destination route is made available. Nodes that are parts of an active route will always offer connectivity
information and power_level by broadcasting periodically local “HELLO” messages to immediate neighbor nodes. If such messages stop arriving from a neighbor beyond fixed time threshold or arrives with power_level<5, the connection is assumed to be lost. The receiving node will remove it from route entry and send a RERR to neighbors of active neighbor lists i.e. nodes using that route. A source that receives an RERR can reinitiate a RREQ message and new route to be introduced based on energy_level. This approach minimizes the long round trip time of basic AODV protocol, more data transfer in a specific time and results reliable and shortest path and small setup delay. It can avoid multiple Route Reply packet and periodic beaconing (active discovery) that was supposed to lead unnecessary bandwidth consumption. Modified AODV will not support unidirectional link.

C. TORA (Temporally Ordered Routing Algorithm)

TORA, a family of link reversal protocols, provides several routes between a source and a destination basically contains 3 parts: creating, maintaining and erasing routes. At each node, a separate copy of TORA is in run per each destination associated a height and builds a directed acyclic graph at the destination. Messages flow from nodes with higher height to those with lower heights. Routes normally are discovered using QRY and UDP packets [20]. Activity Scheduling and route rotation would be perfect one to implement here. When a node with no downstream links needs a route to a destination, it broadcasts a QRY packet with power_level that propagates till it either finds a node with a route to the destination or the destination itself. That node will respond by broadcasting a UPD packet containing the node’s height. Power_level is recommended here to take part in calculation of node’s height. A node receiving the UPD packet updates its height accordingly and broadcast another UPD. This may results a number of directed paths from source to the destination. Obviously higher height nodes are more energized and other nodes can rely on them for communication purposes. For any reason, power_level<5 situation occurs in any node, then new routes are switched automatically with next height accordingly. Again, if a node discovers a particular destination to be unreachable, it sets the corresponding local height to a maximum value. In case the node cannot find any neighbor with finite height with respect to this destination, it attempts to find a new route. Also, in the case where is no route to a destination, the node broadcasts a CLR messages resulting all routing states and removing invalid routes and routes of low energized member nodes from its part of the network. Such modifications can easily avoid non-optimal routes & transient loops, the basic drawbacks of the original algorithm.

IX. CONCLUSION

In this paper, a WCDS based routing algorithm is modified to make it power-aware. This process of dynamic adjustment of backbone nodes results not only prolonged lifetime of a large network but also ensures improved throughput in dense environment where communication is assumed to occur frequently between a large numbers of nodes; the key feature of this algorithm. This approach takes advantages of distributing and efficient using of network resources, reducing network congestion and increasing overall performance. Further investigation should be devoted to assess the algorithm performance in more realistic conditions. It would be also interesting to provide service differentiation with TCP protocols and compare its performance with other MCDS schemes. This work is left open for future.

X. REFERENCES


