CONNECTIVITY BASED EQUIVALENCE PARTITIONING OF NODES TO CONSERVE ENERGY IN MOBILE AD HOC NETWORKS

by

ANAND PRABHU S
ARUN K
RAGHURAM K R

A Thesis Report submitted to the

FACULTY OF INFORMATION AND
COMMUNICATION SYSTEMS

in partial fulfillment of the
requirements for the award
of the degree of

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY
CHENNAI – 600 025

APRIL 2004

BONA FIDE CERTIFICATE

Certified that this thesis titled “Connectivity Based Equivalence Partitioning of Nodes to Conserve Energy in Mobile Ad Hoc Networks” is the bona fide work of

Anand Prabhu S	(20001002)
Arun K	(20001010)
Raghuram K R	(20001062)

who carried out the work under my supervision.

Certified further, that to the best of our knowledge, the work reported herein, does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on these or other candidates.

Dr. Geetha T V,
Head of Department,

Dr. Narayanasamy P,
Professor,

Department o Computer Science and Engineering
College of Engineering Guindy
Anna University
Chennai 600 025

ABSTRACT

This thesis introduces a *connectivity based equivalence partitioning* (CEP) algorithm that reduces energy consumption in Mobile Ad hoc networks. CEP conserves energy by identifying nodes that are equivalent from a routing perspective and then turning off unnecessary nodes, keeping a constant level of routing fidelity. CEP moderates this policy using application and system level information. Nodes that source or sink data remain on and intermediate nodes monitor and balance energy use. CEP is independent of the underlying ad hoc routing protocol. We simulated CEP over unmodified AODV. An analytical model has been developed and it describes the increase in performance of the network when CEP is used along with AODV. We have also simulated CEP in the ns-2 network simulator and the simulation studies show the CEP consumes 40% to 50% less energy when the node density is *high* than an unmodified ad hoc routing protocol viz., AODV or DSR. Moreover, simulations of CEP suggest that the network lifetime increases proportional to node density. CEP is an example of *adaptive fidelity,* a technique proposed for extending the lifetime of self-configuring systems by exploiting redundancy to conserve energy while maintaining fidelity.

ACKNOWLEDGEMENT

We express our heartfelt thanks to our project guide Dr. P. Narayanasamy, Professor, Department of Computer Science and Engineering, Anna University for his support and guidance throughout the course of this project.

We would also like to thank the review panel consisting of Dr.V.Ramachandran, Dr.V.Uma Maheswari, Mrs.D.Manjula, Mr.P.Yogesh and Mrs.T.K.S.Lakshmi Priya, esteemed faculty of the Department of Computer Science and Engineering, Anna University for having provided constant guidance and help to complete our work successfully.

Finally, we would like to express our heartfelt thanks to all other esteemed faculty members who have imparted an engineering education to us, our family members and student friends who have contributed directly and indirectly to the success of this project.

ANAND PRABHU S
ARUN K
RAGHURAM K R

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Mobile hosts and wireless networking hardware are becoming widely available, and extensive work has been done recently in integrating these elements into traditional networks such as the Internet. Oftentimes, however, mobile users will want to communicate in situations in which no fixed wired infrastructure such as this is available, either because it may not be economically practical or physically possible to provide the necessary infrastructure or because the expediency of the situation does not permit its installation. For example, a class of students may need to interact during a lecture, friends or business associates may run into each other in an airport terminal and wish to share files, or a group of emergency rescue workers may need to be quickly deployed after an earthquake or flood. In such situations, a collection of mobile hosts with wireless network interfaces may form a temporary network without the aid of any established infrastructure or centralized administration. This type of wireless network is known as a *mobile ad hoc network*.

In ad hoc networks all nodes are mobile and can be connected dynamically in an arbitrary manner. All nodes of these networks behave as routers and take part in discovery and maintenance of routes to other nodes in the network. Ad hoc networks are very useful in emergency search-and-rescue operations, meetings or conventions in which persons wish to quickly share information, and data acquisition operations in inhospitable terrain.

The rest of this chapter is devoted to an introduction to mobile ad hoc networks and the related work in this area.

1.1 MOBILE ADHOC NETWORKS

In Figure 1.1, we see three wireless mobile hosts that are in a mobile ad hoc network. Node B is within the transmission range of node A. Both nodes A and C are in the transmission range of node B. Node B is in the transmission range of node C.

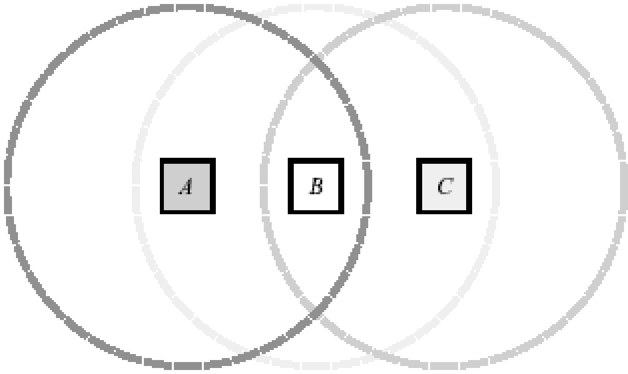


Figure 1.1: A simple ad hoc network of three wireless mobile hosts

We see that node A is not directly connected to node C. So, in order to route information from node A to C or vice versa, node B is needed as an intermediate node. This is a simple example of a multihop mobile ad hoc network. In practical ad hoc networks, routing becomes much more complicated and lots of decisions needs to be taken in a distributed manner for the proper functioning of the network.

1.2 CURRENT METRICS

A number of routing protocols have been proposed to provide multi-hop communication in wireless, ad hoc networks [1, 2, 3, 4]. Traditionally these protocols are evaluated in terms of packet loss rates, routing message overhead, and route length [5, 6, 7]. Since ad hoc networks will often be deployed using battery-powered nodes, comparison and optimization of protocol energy consumption is also important.

When ad hoc networks are deployed using battery-powered nodes, the important question of how limited energy resources affects system lifetime and overall performance becomes critical. For scenarios such as sensor networks where energy maps directly to lifetime and utility, energy use is ***the*** important metric. To understand energy efficiency we examined existing ad hoc routing protocols.

On-demand protocols, by their very nature are more efficient in the energy consumed by routing overhead packets. As a result, energy use is dominated by routing protocols overhead. In fact, the major source of extraneous energy consumption was from overhearing. Radio has a relatively large broadcast range. All nodes in that range must receive each packet to determine if it is to be forwarded or received locally. Although most of these packets are immediately discarded, they consume energy with this simple energy model. This observation motivates approaches that avoid overhearing.

1.3 MOTIVATION

Actual radio consumes power not only when sending and receiving, but also when *listening* or *idle* (the radio electronics must be powered and decoding to detect the presence of an incoming packet). Research [8, 9] shows that idle energy dissipation can not be ignored in comparing to sending and receiving energy dissipation. Stemm and Katz show idle : receive : transmit ratios are 1 : 1.05 : 1.4 by measurement [8], while more recent studies show ratios of 1 : 2 : 2.5 [9], and 1 : 1.2 : 1.7 [10]. In any of these cases, energy dissipation in idle state can not ignored. With such energy model, all ad hoc routing protocols considered consume roughly the same amount of energy (within a few percent) as shown in the Figure 1.2.

In the scenario with modest traffic, idle time completely dominates system energy consumption. The studies based on an energy model that considers energy dissipation in sent/received packets and idle time, suggests that energy optimizations must turn off the radio, not simply reduces packet transmission and reception. Powering off radio conserves energy both in overhearing due to data transfer and in idle state energy dissipation when no traffic exists. We therefore explore nodes that power down their radios much of the time. This approach is similar to the use of TDMA for power conservation [11], or PAMAS [12]. However, unlike these approaches, we employ information from above the MAC-layer to control radio power. The application or routing layer provides better information about when the radio is not needed.

On the other hand, we observed that when there is significant node redundancy in an adhoc network, multiple paths exist between nodes. Thus we can power off some intermediate nodes while still maintaining connectivity. For example in Figure 1.2, if node 2 is awake, node 3 and 4 are extraneous for the communication between 1 and 5.

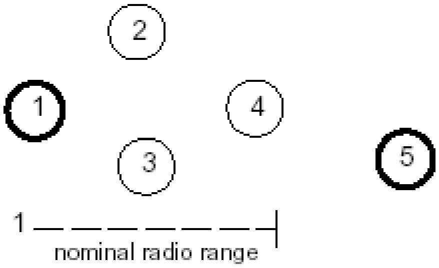


Figure 1.2: Example of node redundancy in ad hoc routing

We define *routing fidelity* as uninterrupted connectivity between communicating nodes. Thus routing fidelity can be maintained as long as any intermediate node is awake.

1.4 RELATED WORKS

Reducing energy consumption has been the recent focus of wireless adhoc network research. The Geographic Adaptive Fidelity (GAF) [13] scheme of Xu et al. self configures redundant nodes into small groups based on their geographic locations and uses a localized, distributed algorithm to control node duty cycle to extend network operational lifetime. But in many settings, such as indoors or under trees where GPS does not work, location information is not available. The dependency on global location limits GAF's usefulness. In addition, geographic proximity does not always leads to network connectivity.

The SPAN [10] scheme of Chen and Jamieson proposes a distributed algorithm for approximating connected dominating sets in an adhoc network that also appears to preserve connectivity. SPAN elects coordinators by actively preventing redundant nodes by using randomized slotting and damping. Equivalence partitioning differs from GAF as it constructs the partitions based on the connectivity information rather than the geographic location of the nodes. Also unlike SPAN, it constructs equivalence partitions and randomly rotates the active nodes within the partition.

1.5 ORGANIZATION OF THE THESIS

The thesis is organized as follows: Chapter 2 explains the need and characteristics of a Topology Maintenance Algorithm. Chapter 3 deals with the proposed Connectivity Based Equivalence Partitioning Algorithm. Chapter 4 gives the details about the extensive simulation carried out and discusses the results. Chapter 5 concludes the thesis with suggested future work.

CHAPTER 2
TOPOLOGY MAINTENANCE PROTOCOLS

2.1 WHY TOPOLOGY MAINTENANCE?

Since ad hoc networks will often be composed of battery-powered nodes, energy consumption is also an important metric. When a node is not participating in routing, forwarding, sending or receiving it is in the idle state. The node is dissipating power even when it is in the idle state. The idle to off state power ratio can be defined as:

$$P_{idle} / P_{off}$$

When this ratio is more, lot of power is wasted when the node is idle. Since the network interface may be often idle, power could be saved by turning off the radio when not in use. But the coordination of power saving with routing in adhoc wireless networks is not straight forward.

Here we present a topology maintenance algorithm which partitions the network in such a way that one on the nodes in each partition must be active so that the connectivity of the network does not diminish and the other nodes can turn off their radio. The responsibility of the active node is randomly changed so that every node is treated equally and the life time of the over all network is increased.

2.2 CHARACTERISTICS OF AN IDEAL TOPOLOGY MAINTENACE ALGORITHM

A good power-saving coordination technique for wireless ad-hoc networks ought to have the following characteristics. It should allow as many nodes as possible to turn their radio receivers off most of the time, since even an idle receive circuit can consume almost as much energy as an active transmitter. On the other hand, it should forward packets between any source and destination with minimally more delay than if all nodes were awake. This implies that enough nodes must stay awake to form a connected backbone. The algorithm for picking backbone should be distributed, requiring each node to make local decision. Furthermore, the backbone formed by the active nodes should provide about as much total capacity as the original network, since otherwise congestion may increase. This means that paths that could operate without interference in the original network should be represented in the backbone.

For example, Figure 2.1 illustrates a topology that violates this principle. In this topology, black nodes are coordinators. Nodes that are within the radio range of each other are connected by solid or dotted lines.

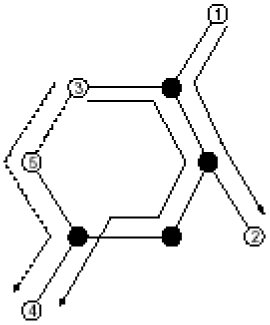


Figure 2.1: A simple topology

Packets between nodes 3 and 4 may contend for bandwidth with packets between nodes 1 and 2 (solid arrow). On the other hand, if node 5 was a coordinator, nodes 3 can send packets to node 4 through node 5 via the path shown by the dotted arrow and thus no contention would occur.

A good coordination technique should not make many assumptions about the link layer's facilities for sleeping; it should work with any link-layer that provides for sleeping and periodic polling, including 802.11 ad hoc power saving mode. Finally, power saving should inter-operate with what ever routing system the ad hoc network uses.

Thus in our design of the connectivity based equivalence partitioning algorithm we have taken care that maximum number of nodes go to sleep state maintaining the connectivity of the network in a stable state. This algorithm also does not depend on the underlying routing algorithm or the link layer algorithm as it uses application- and system-level information.

CHAPTER 3

CONNECTIVITY BASED EQUIVALENCE PARTITIONING ALGORITHM

3.1 INTRODUCTION

We now present our energy conserving topology maintenance algorithm *Connectivity Based Equivalence Partitioning* (CEP). Each CEP node uses its adjacency information to associate itself with a 'partition', where all nodes within a particular partition are equivalent with respect to forwarding packets. Nodes in the same partition then coordinate with each other to determine who will sleep and how long; this determination is moderated by application and system information. Nodes then periodically wake up and trade places to accomplish load balancing. We also consider how CEP interacts with the underlying ad hoc routing protocol.

3.2 EQUIVALENCE PARTITIONING ALGORITHM [15]

This Equivalence Partitioning algorithm is as follows.

- The node N_i constructs its neighbor set by sending HELLO packets to its one hop neighbors. The nodes hearing this packet responds with a HELLO reply so that the node Ni constructs its neighbor set. Let NH_i be the neighbor set of node N_i
- The node N_i advertises its neighbor set to its one hop neighbors so that it can find out the number of pairs of its neighboring nodes connected via this node.
- Find the intersection between the neighbor sets of the adjacent nodes. Let C be the cardinality of the intersection set with the first neighbor.
- If the cardinality is equal to or more than two then form equivalence partitioning and assign a unique partition id to the nodes.

- Consider the next neighbor. Let C' be the cardinality of the intersection set between the node Ni and its neighbor currently considered. If C' > C, a new group is formed between the node Ni and this neighbor, destroying the previous partition.
- If C' = C with same elements then add the new neighbor to the same partition and assign the partition id.
- Repeat the above process until each node receives the neighbor set from all its one hop neighbors.

By this way, we construct Equivalence Partitions among the nodes in the Mobile network. Each and every node is exactly in one of the partitions.

The Figure 3.1 shows a hypothetical network of five nodes. The node A and E are adjacent to B, C and D. Node B and D are adjacent to A, C, and E. The node C is adjacent to A, B, C and D.

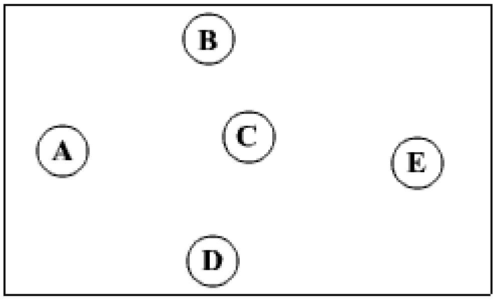


Figure 3.1: A Hypothetical Network with five nodes

In Figure 3.2 we find the neighbor sets of each node shown. The nodes advertise their neighbor sets to their one hop neighbors **only**.

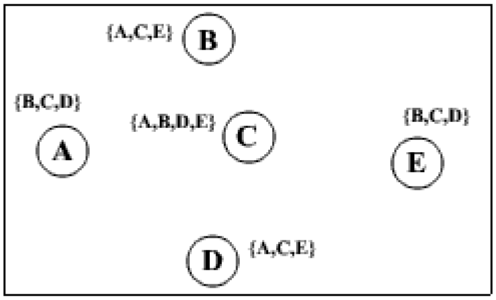


Figure 3.2: The nodes with their Neighbor Sets

In this example, we assume that B advertises its neighbor set first to A. The intersection set is {C}. So we do not form a partition. When B advertises its neighbor set {A, C, E} to C then the intersection set is {A, E}. Now we form a partition with the nodes B and C. When D advertises its neighbor set to C the intersection set is {A, E}. So we add D to the partition containing B and C. Thus a partition containing B, C and D is formed.

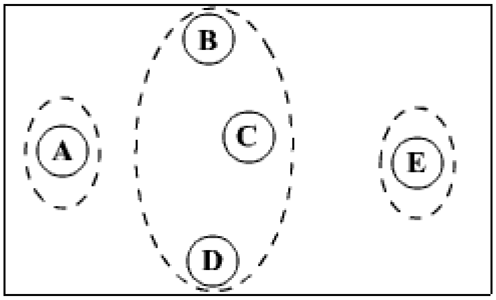


Figure 3.3: The *Equivalence Partitions* shown by dotted lines

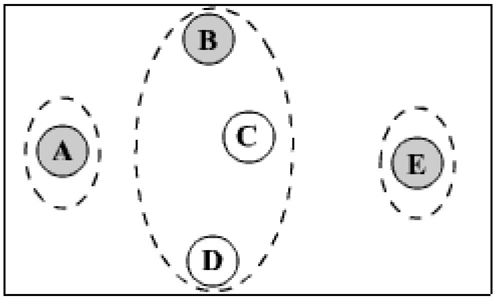


Figure 3.4: The Active nodes shown by the Grey Circles

When C advertises its neighbor set to A the intersection set is {B, D} and the cardinality is two. But the elements in the neighbor set are different. So we do not form a partition consisting of A and C. Even though A and E have their neighbor set as B, C and D they cannot form a partition because they are not one hop neighbors. Thus we find that the Equivalence Partitions formed as shown in the Figure 3.3. The equivalence partitions may vary depending on the order of advertisement of the neighbor sets by the nodes. In Figure 3.4 we find that the active nodes selected in each Equivalence Partition. The nodes with the least id in each partition are chosen to be active for the first time. The least id of the node in a partition is assigned as the partition id for all the nodes in that partition.

3.3 CEP STATE TRANSITION

The overall System Design of the connectivity based equivalence partitioning algorithm can be described using the State Diagram shown in Figure 3.5.

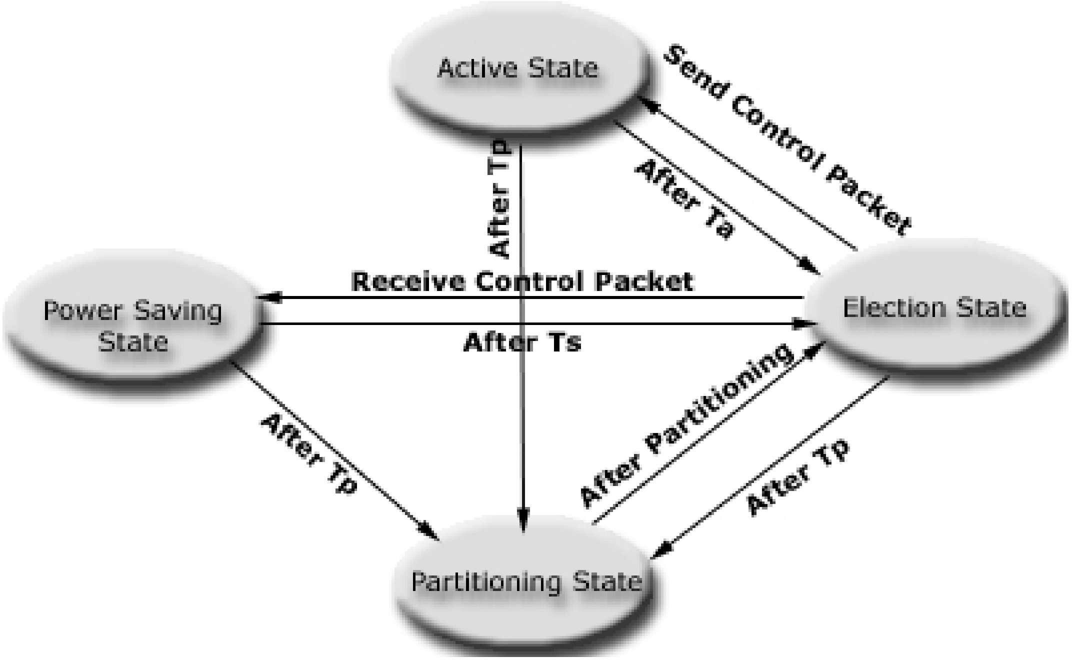


Figure 3.5: State Transition in CEP

When we start with a new network, all the nodes are in the *Partitioning State*. In the Partitioning State, each node send Hello Packet {node id, ttl=1} to its one hop neighbors and waits for a period so that it receives HELLO REPLY packets from its one hop neighbors. Once it receives the HELLO REPLY packets from all its neighbors it forms the neighbor sets. Each node exchanges its neighbor set and calculates the intersection to form the equivalence partitions.

After forming the partitions, each node is assigned the *partition id* and the ids of the other nodes in its partition. The partition id is the id of the node with the least node id in the partition. A node now knows what the other nodes in its own partition are and uses the least id of the node in the partition as the partition id. This ends the *Partitioning State* and the node moves to the *Election State*.

In the *Election State*, each node waits for a random amount of time say, δ which is inversely proportional to the remaining power of the nodes and directly proportional to the node ids. Thus nodes with high power need to wait for a small delay and vice versa. Also when we start with a new network all nodes will have the same power and so we need to back off based on the ids of the nodes in the partition. When the nodes have equal power, the node with the least id gets a chance to wait for the minimum delay. Once a node waits for the delay δ, it sends a control packet which consists of the node id, estimated network active time (*enat*) to all the other nodes. Thus we see that the node with the maximum remaining power and least id send the control packet first and enters into the *active state*. We can have the estimated network active time *enat* equal to *enlt/2* where *enlt* id the Estimated Network life time. The node remains in the active state for a time Ta which is equal to the estimated network active time *enat*.

The nodes receiving the control packets have lesser energy than the node which sent the control packet. Thus after receiving the control packet, all the other nodes in the partition go to the *Power Saving State*. They remain in the power saving state for a time Ts which is chosen in random from the interval [*enat/2*, *enat*]. By this way the other nodes sleep for a time less than or equal to the active time of the active node. After Ts, the nodes in the *Power Saving State* move to the *Election State*. The node in the active state also moves to the *Election State* after Ta and the process of election of the new active node starts. After time Tp which is dependent on the mobility of the network, the nodes in any state will come to the *Partitioning State* and the partitioning algorithm repeats.

3.4 TUNING THE PARAMETERS OF CEP

CEP leaves choice of many parameters including enat, Tp, Ta, Ts, δ to the application. In this section we describe how and why these parameters are chosen in the current CEP Algorithm. Applications may wish to optimize these choices, for example, perhaps trading increased packet loss for greater energy savings.

Estimated node active time (*enat*) can be set to the expected node lifetime (*enlt*), conservatively set by assuming the node will constantly consume energy at a minimum rate until it dies. Rather than this conservative *enat*, CEP uses an approach described in section 3.5 to balance energy usage across nodes.

CEP selects the value of δ as a uniform random number between zero and a conant value. This delay δ is defines As follows

```
δ = (Random::uniform(DELTA_CONSTANT-1,DELTA_CONSTANT) +  50*nid_)/currentEnergy;
```

where nid is the id of the node, *currentEnergy* is the remaining energy of the node.

So the node having the least node id and the node with the maximum power waits for the minimum delay.

Nodes active duration (T_a) can be its expected lifetime (enlt). CEP instead uses T_a to accomplish load balancing as described in section 3.5

Node sleep time (Ts) can be set to the enat of the active node since this is the conservative assumption of its lifetime. Due to node mobility, the active node may move out the partition. This can leave a partition without any active nodes although some nodes are sleeping, reducing routing fidelity. One approach that statistically reduces this problem is to set Ts as a uniform random time between 0 and enat. This large range of Ts may often have nodes wake up quite early. In CEP therefore Ts is uniformly from the range [enalt/2, enalt].

3.5 LOAD BALANCING ENERGY USAGE

CEP employs a load balancing strategy so that all nodes remain up and running together for as long as possible. The idea behind this is that all nodes in the network are equally important and no one node must be penalized more that any other node.

CEP uses the following load balancing strategy. A node sends a control packet only when it does not receive a control packet from any other node in the partition. We have designed the control packet transfer mechanism such that the node with the minimum node id and maximum power send first. So any node receiving a control packet goes to the sleep state and the node sending the control packet goes to the active state.

After a node remains in the active state for the time Ta, it changes its state to the *election state* to give a chance to other nodes within the same partition to become active. When the active node changes to the election state, it is more likely to that it has less remaining energy that its neighbors because presumably the neighbors were in the sleep state conserving energy during the node's active time. Consequently, the node that was active is less likely to remain active after the next election phase.

The active node sets Ta to the value of enat and advertises enat in its control messages. The non-active nodes in the neighborhood use enat to determine their sleeping period. The active node sets enat to a value less than the time to use up all remaining energy (*enlt*). In our simulation we set enat to enlt/2 so that node consumes half of its energy before handing off to another node in the neighborhood.

3.6 ADAPTING TO HIGH MOBILITY

CEP tries to adapt the number of nodes participating in ad hoc routing to keep a constant level of nodes that route data. The ideal scenario would be one active node in each partition at a time. However, as nodes move, the active node may leave the partition. This may leave the prior partition without an active node, reducing routing fidelity. In scenarios with high mobility this problem can greatly increase packet drop rates.

We can accommodate high mobility by considering this system-level behavior explicitly in CEP. Each node estimates the time it expects to leave the partition *enpt*. When other nodes enter sleeping state, they sleep for the smaller of *enat* and *enpt* to decide how long it can stay in the sleep state.

3.7 CEP INTERACTION WITH AD HOC ROUTING

In principle, CEP will run over any ad hoc routing protocols because it only uses application- and system-level information to decide each node's duty cycle, and since discovery message are only broadcast to direct neighbors.

CEP decision to turn nodes on and off is independent of ad hoc routing protocols. If a node is actively routing packets when it is powered off, CEP depends on the routing protocol quickly re-routing traffic. This may cause some packet loss, although most ad hoc routing protocols react to changes quickly. An optimization that we have not explored is to have CEP inform the ad hoc routing protocol of impending suspension, allowing it to preemptively re-route any traffic.

CHAPTER 4

PERFORMANCE ANALYSIS

To evaluate our scheme, we first use a simple mathematical analysis to determine an idealized level of energy conservation in CEP. Since a mathematical analysis cannot capture the complexity in a full CEP scenario, we then use simulation to study CEP effects on network lifetime, the average energy or power of the network nodes under CEP. All these results are compared with AODV i.e. CEP over AODV is compared with unmodified AODV. We also study whether or not it increases the packet drops in the network. Finally we show that the network lifetime under CEP is proportional to the density of the node deployment.

4.1 ANALYTICAL PERFORMANCE ANALYSIS

To get an upper bound on how much CEP may extend network lifetime we next consider a very simple analytic model. Assume that *n* nodes are evenly distributed in a area with topography size A. Let the nominal radio range be *R*. The grid size of a partition can be set as $\pi * R^2$ which is the maximum size of a virtual partition. Therefore the minimum number of virtual partitions *p* would be

$$p = A / (\pi * R^2) \qquad (1)$$

According to our assumption of evenly distributed nodes, each partition would have at most n/p nodes, or

$$n * \pi * R^2 / A \qquad (2)$$

nodes.

At best, assuming stationary nodes and no CEP overhead, only one node in each partition will be active while the rest sleep. Since equation (2) gives the maximum number of nodes in each partition, the network lifetime will be extended at most $(n * \pi * R^2 / A)$ times.

The formula basically reflects the fact that with CEP algorithm, the more nodes, longer is the network lifetime and the fewer number of partitions, longer is the network lifetime. The number of virtual partitions mainly depends on the nominal radio transmission range and the topography size.

Although CEP periodically sends out various messages like control packets, hello messages this frequency will be very low. Also since the broadcast is limited to just the one hop neighbors, such overhead will not affect the whole system energy dissipation too much.

In the following sections, we use simulation to relax our assumptions and explore CEP performance in more realistic conditions.

4.2 SIMULATION METHODOLGY

Since it is difficult to capture the details of CEP performance in an analytical model, we have implemented CEP in the ns-2 simulator, evaluating variations in node movement, traffic pattern and energy model as described below.

In order to demonstrate the flexibility of CEP, we implemented CEP in a snapshot of ns-2.1b8. We used locally modified and extended version of adhoc routing contributed by CMU[14], and an energy model described below. We attached CEP to AODV to get CEP/AODV, and CEP to DSR to get CEP/DSR. We then run CEP/AODV, AODV, CEP/DSR and DSR on the same simulated scenarios to compare the performance in terms of the energy dissipation, average network lifetime and the data delivery quality.

4.3 TRAFFIC AND MOBILITY MODELS

Nodes in the simulation move according to the random way-point model used in CMU[5]. Nodes alternate between passing and then move to a randomly chosen location at a fixed speed. We consider many pause times during the simulation time period. For each pause time we generate 10 sets of initial placements and random way-points. We also evaluate two different node movement speeds: uniform distribution between 0 and 20 m/s and uniform distribution between 0 and 1 m/s. Nodes move in a 1500m by 300m meter area.

We also study the effect of node density on CEP by varying the number of nodes from 10 to 100, while keeping the area constant. In most scenarios we use

10 traffic nodes and other nodes as transit nodes. Simulation traffic was generated by continuous bit rate (CBR) sources spreading the traffic randomly among 10 traffic nodes. The packet size was set to three different values: 1 pkt /s, 10 pkts/s and 20 pkts/s to evaluate CEP sensitivity to traffic load. Even with 20 pkts/s packet rate, the traffic node is still well below the network capacity. We chose to study light to moderately loaded systems because nodes in ad hoc networks are expected to be energy constrained, more than bandwidth constrained. We model a radio with a nominal range of 250 meters.

4.4 ENERGY MODEL

Our energy consumption model is based on Stemm and Katz's measurements of a 1995 AT&T 2 Mb/s Wave LAN (pre-802.11) wireless LAN [8]. They measured costs of 1.6W for transmit, 1.2W for receiving, and 1.0W for listening. To this we add a cost of 0.025W when sleeping. We chose their model as representative at the time we began this work. Although this hardware is now somewhat old, newer evaluations of more recent versions of the Wave LAN card, and compatible hardware by other vendors shows very similar costs.

It is impossible to evaluate the behavior of the network as whole if the traffic nodes run out of energy before the transit nodes. To avoid this, we separate the traffic nodes from routing nodes and give traffic nodes infinite energy. Traffic nodes follow the same mobility model as transit nodes, but they do not run CEP, nor do they forward traffic. Because we treat traffic nodes specially, we do not count them when reporting the number of nodes in the simulation (for example, our 10 node simulation consists of 20 nodes- 10 transit nodes and 10 traffic nodes).

We give each transit node enough energy so that it can listen for about 500 seconds. Since nodes in AODV and DSR have nodes listening constantly, all nodes expire after 450s even without traffic.

We compared CEP/AODV with normal AODV and CEP/DSR with normal DSR. The remainder of this section presents our simulation results. The sections 4.5-4.6 analyze CEP performance in terms of network lifetime, energy conservation, and data delivery quality under low mobility with various traffic loads. We have evaluated both AODV and DSR for all nodes and found similar behavior. So we have reported only the comparison between CEP and AODV in the report that follows.

4.5 CEP ENERGY SAVINGS AND NETWORK LIFETIME

In order to quantify how much energy CEP saves, we compute the mean energy consumption per node. To define the mean energy consumption per node, we assume that simulation starts with *n* nodes with initial total energy of *n* nodes as E_0. After time *t*, the remaining total energy of *n* nodes is E_t. Then the mean energy consumption per node (*aen*) is**:**

$$aen = (E_0 - E_t)/(n * t)$$

We calculate *aen* for both AODV and CEP over AODV for all different scenarios including movement pattern and different traffic load.

Figure 4.1 shows the graph of average energy consumption for both CEP and AODV at low node speed (1m/s), movement 1m/s and traffic 20 pkts/s and simulation time of 500 seconds.

From the Figure 4.1, it is clear that even for 10 nodes, the energy consumed is saved up to 16%. The figure 4.2 below shows the network lifetime graph. It is clear from the graph that CEP does increase the network lifetime. After 400 seconds, all nodes under AODV lose their energy where as in CEP, 50% of the nodes survive. Even after 500 seconds, 30% of the nodes manage to survive.

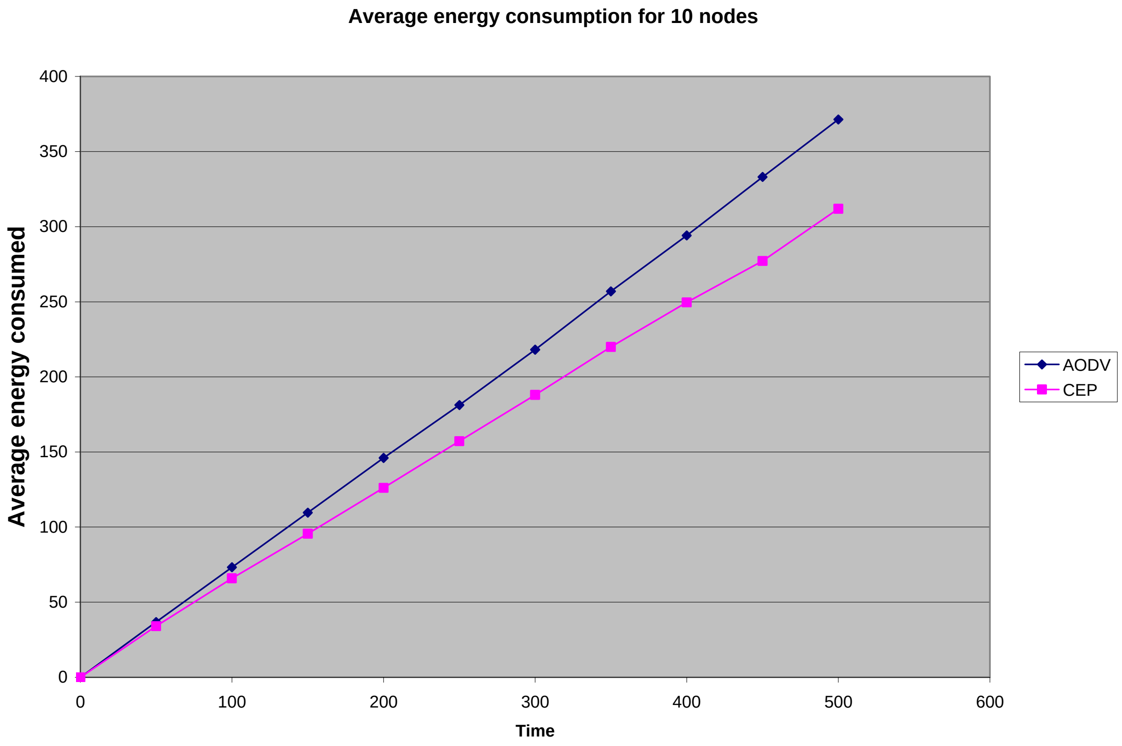


Figure 4.1: Average energy consumption for 10 nodes.

This difference can be explained from the fact that AODV does nothing to conserve energy. This represents the cost of continuous listening. In CEP, the lifetime will be enhanced when there is more than 1 node in a partition, for it helps in load balancing. If a partition has only one node, then node will be active always and will soon die.

Figure 4.2 shows the fraction of nodes survived for 10 nodes for both CEP and AODV, for a simulation time of 500 seconds, movement of 1m/s and traffic of 20 pkts/s. It is clear from the graph that even after 400s, some nodes survive under CEP whereas all nodes under AODV die before 400s. This is again due to the cost of continuous hearing.

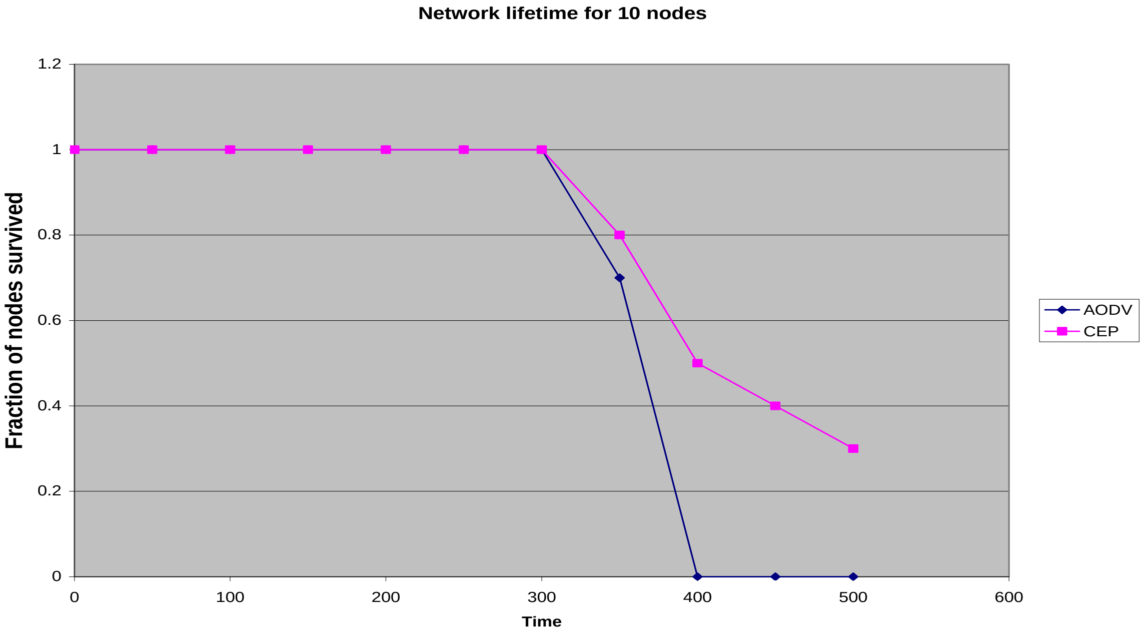


Figure 4.2: Network lifetime for 10 nodes.

We then show the results for 30, 60 and 100 nodes. The graphs are shown below. Figure 4.3 shows the average energy consumption for 30 nodes for both CEP and AODV at low node speed (1m/s), movement 1m/s and traffic 20 pkts/s and simulation time of 500 seconds.

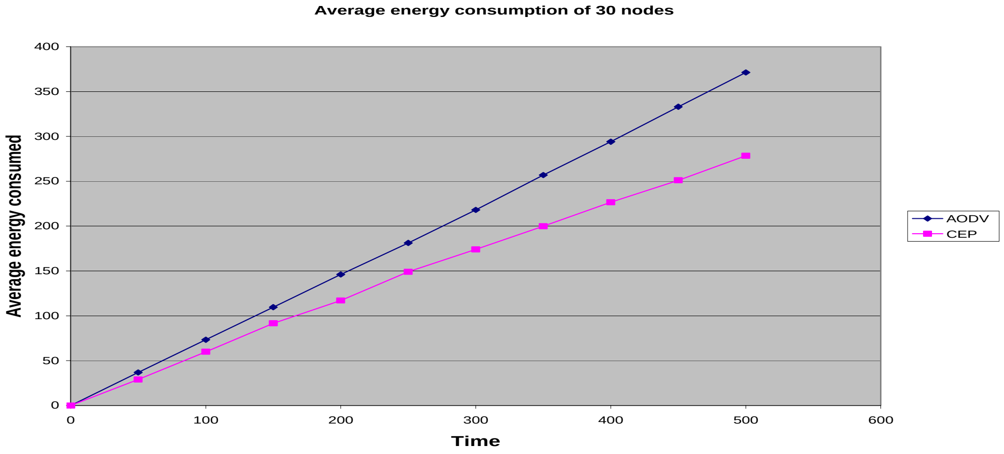


Figure 4.3: Average energy consumption for 30 nodes.

Figure 4.4 shows the fraction of nodes survived for 30 nodes for both CEP and AODV, for a simulation time of 500 seconds, movement of 1m/s and traffic of 20 pkts/s.

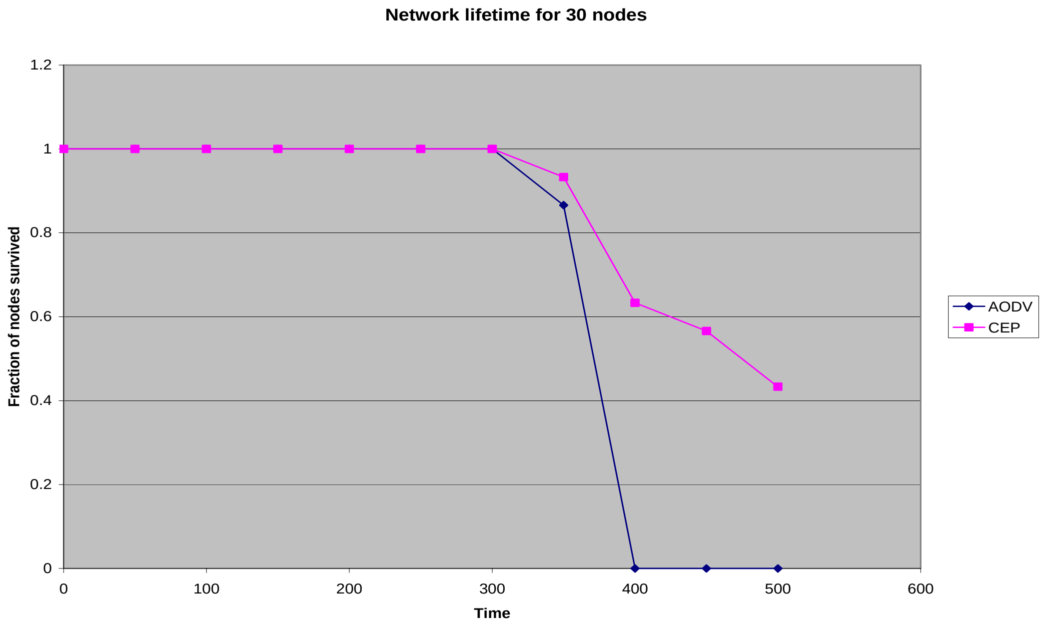


Figure 4.4: The network lifetime for 30 nodes.

Figure 4.5 shows the average energy consumption for 60 nodes for both CEP and AODV at low node speed (1m/s), movement 1m/s and traffic 20 pkts/s and simulation time of 500 seconds.

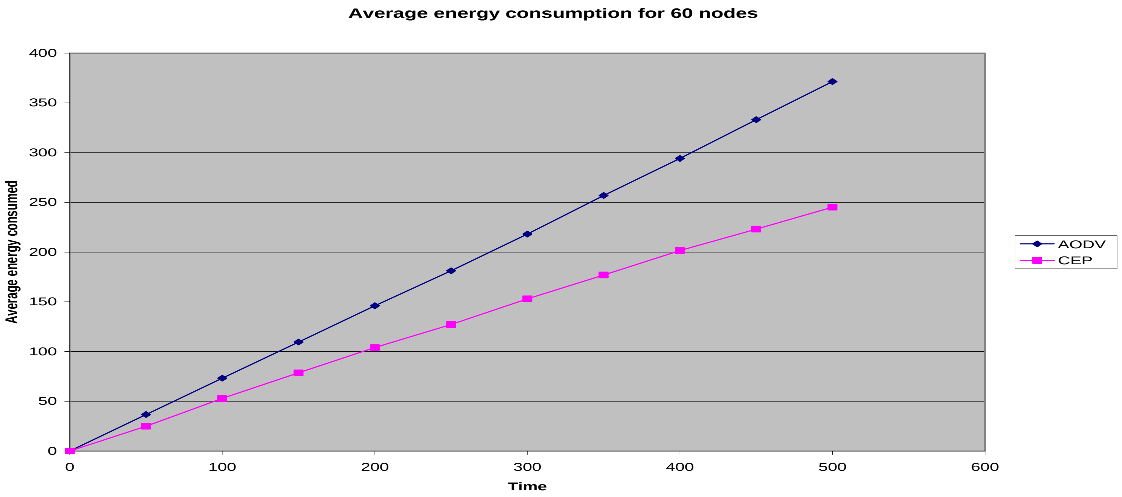


Figure 4.5: The average energy consumption for 60 nodes.

Figure 4.6 shows the fraction of nodes survived for 60 nodes for both CEP and AODV, for a simulation time of 500 seconds, movement of 1m/s and traffic of 20 pkts/s.

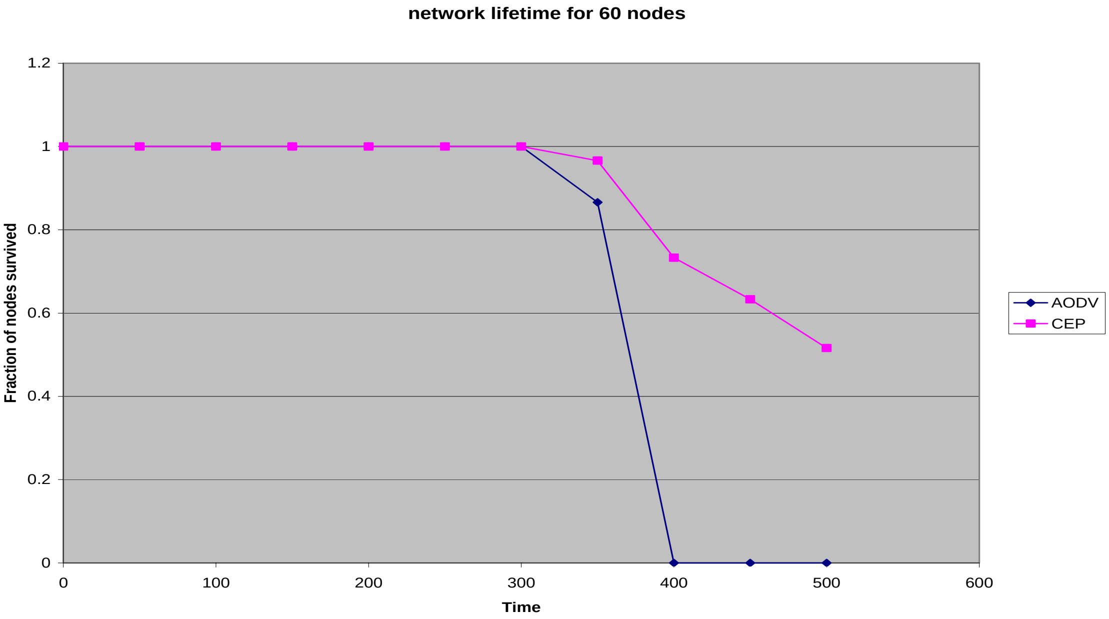


Figure 4.6: The network lifetime for 60 nodes.

Figure 4.7 shows the average energy consumption for 100 nodes for both CEP and AODV at low node speed (1m/s), movement 1m/s and traffic 20 pkts/s and simulation time of 500 seconds.

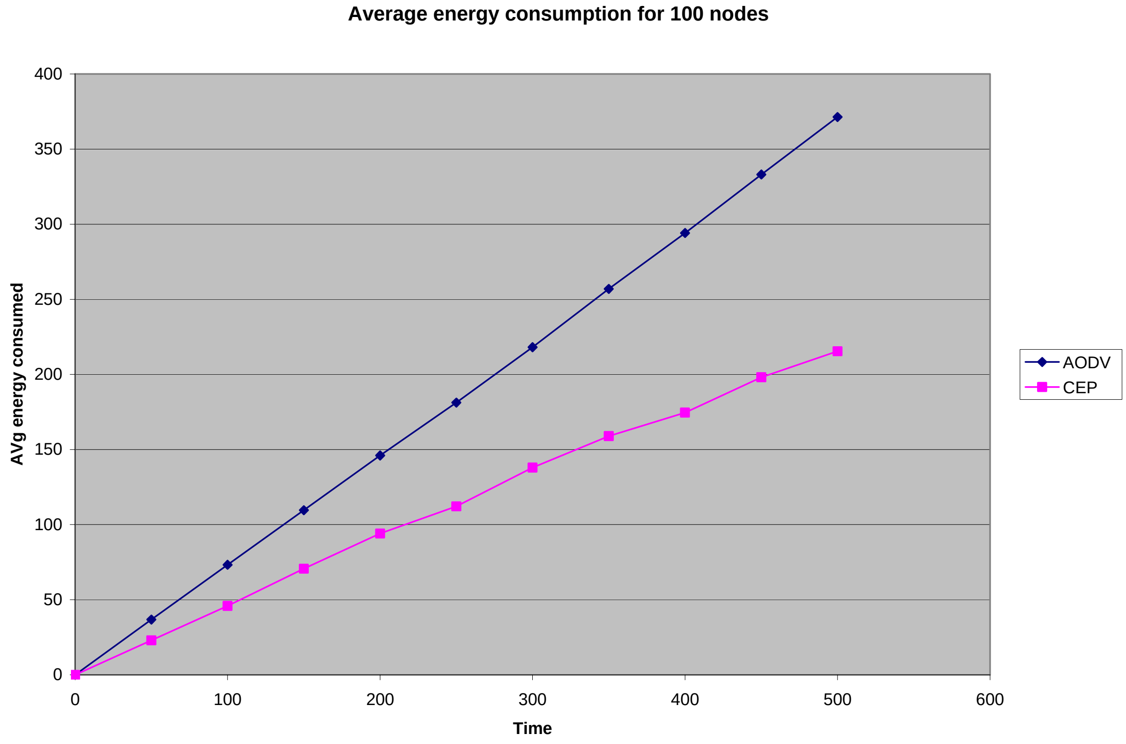
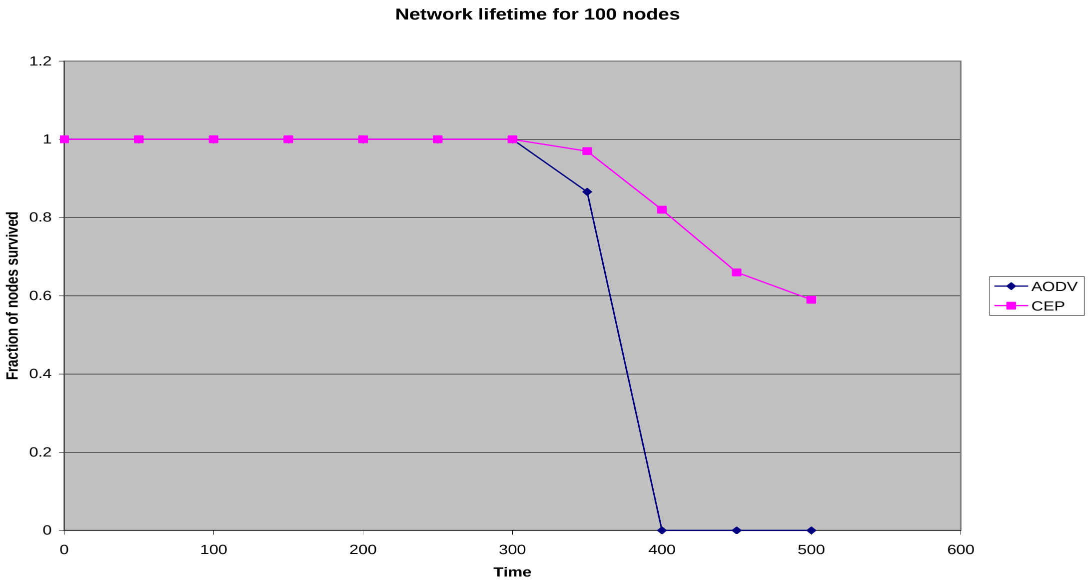


Figure 4.7: The average energy consumption for 100 nodes.

Figure 4.8 shows the fraction of nodes survived for 100 nodes for both CEP and AODV, for a simulation time of 500 seconds, movement of 1m/s and traffic of 20 pkts/s.



.

Figure 4.8: The network lifetime for 100 nodes.

From the above graphs, it is clear that as the node density increases, the average energy consumed decreases, and hence the network lifetime increases. The percentage improvement for 10 nodes is about 16%. But as the number of nodes increases, the percentage improvement goes up. It is 25% for 30 nodes, a significant improvement from that of 10 nodes. When the number of nodes is increased to 60, the energy consumed is reduced as much as 34% and when the number of nodes is 100, 42% of power is saved. Thus CEP works increasingly well with increasing node density.

It is also clear from the graph, CEP increases network lifetime considerably. After 400s all nodes in AODV are unable to survive. But in CEP, at least 40% of the nodes are able to survive when the number of nodes is 30, and the survival rate exceeds 50% as the number of nodes is increased to 100.

4.6 CEP EFFECTS ON DATA DELIVERY

While it is now clear that CEP saves energy, because CEP saves nodes to sleep it may reduce the number of packets that are successfully delivered, thus reducing routing fidelity. We have designed CEP to maintain a constant level of routing fidelity, but we expect some data loss when nodes go to sleep as routes change, and we have identified several cases where node dynamics may cause unintentional data loss.

We define two metrics to measure data delivery. The first metric is the data delivery ratio, which is the ratio of the number of received packets over the total sent packets. The second metric is the average data transfer delay, which is the mean delay for those received packets.

In this section, we compare CEP and AODV during the first 350s of simulation when all the AODV nodes are alive. Our goal is to show that CEP is not significantly worse than AODV when all AODV is effective.

Figure 4.9 shows the data delivery ratio for both AODV and CEP for 100 nodes for a simulation time of 500 seconds, movement of 1m/s and traffic of 20 pkts/s.

Data delivery ratio = (#data packets received)/ (#data packets sent)

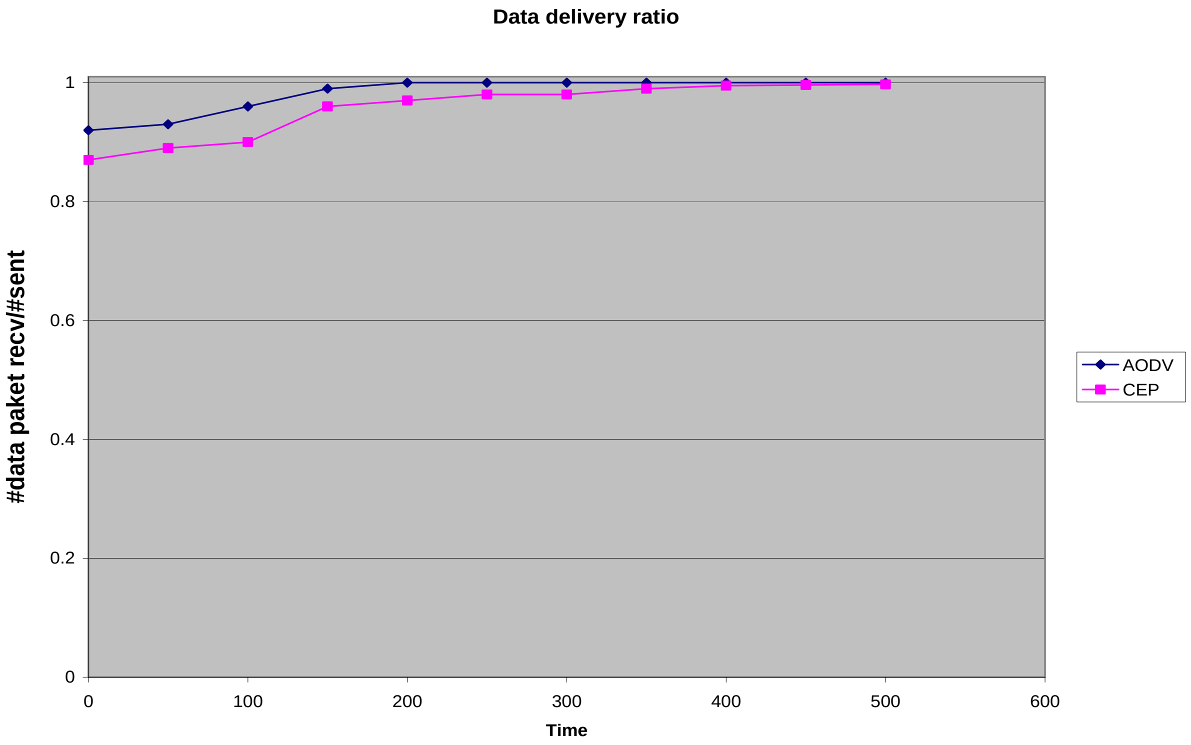


Figure 4.9: Data delivery ratio for 100 nodes.

It is seen that CEP has a very good delivery ratio in a high density network. It is also seen from the graph that CEP does not significantly affect the data delivery ratio in comparison with AODV.

Next we consider the average delay i.e. the average packet delivery delay. Figure 4.10 shows the average packet delivery delay for 100 nodes for both CEP and AODV, for a simulation time of 500 seconds, movement of 1m/s and traffic of 20 pkts/s.

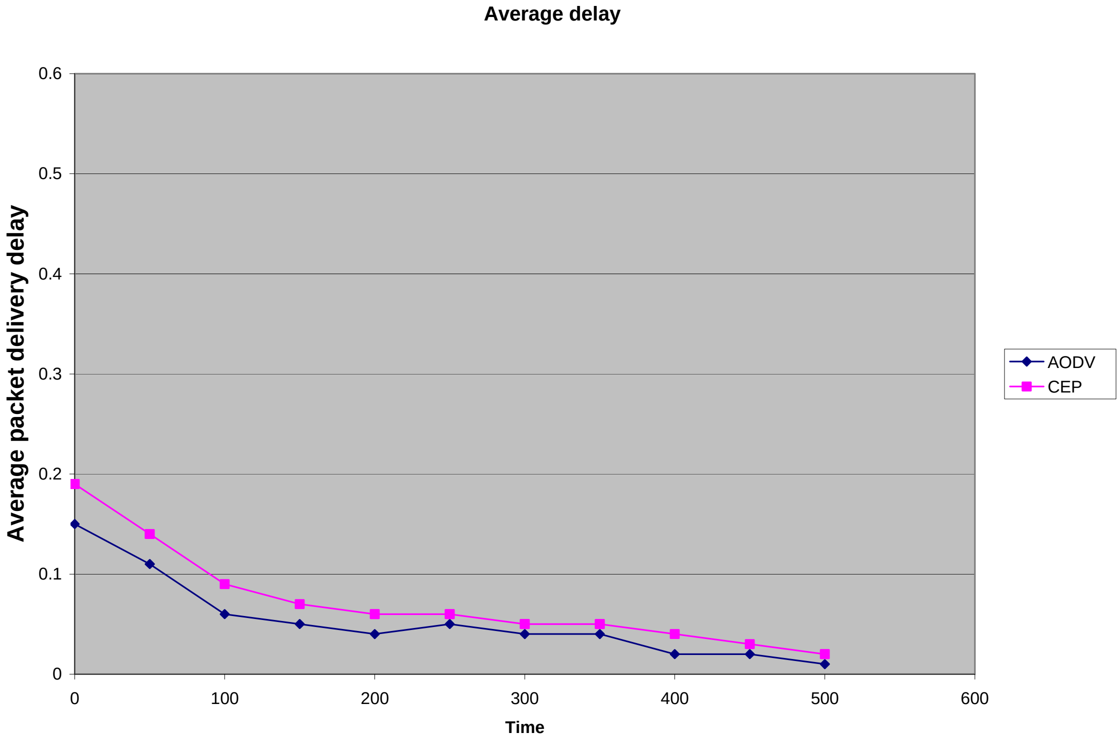


Figure 4.10: The average packet delivery delay for 100 nodes.

From the graph (figure 4.10), it is clear that there is no significant difference in the average packet delay as well.

4.7 CONCLUSION

It is seen from the graphs (figures 4.1, 4.3, 4.5, 4.7) that as the node density increases, the average energy consumption decreases. The percentage savings in energy consumed increases as the node density increases. It is seen from the graph that there is 42% less energy consumption when the number of nodes is 100.

It is also clear from the graphs (figures 4.2, 4.4, 4.6, 4.8) that the network lifetime also increases as the node density increases. For 100 nodes, 60% of the nodes survive even after 500s under CEP, whereas all nodes under AODV lose their power even before 400s.

It is also clear from the graphs (figures 4.9, 4.10) that CEP does not significantly affect the quality of data transmission as well in comparison with AODV.

Thus CEP extends the network lifetime and decreases the average energy consumed without significantly affecting the data delivery ratio and the average packet delivery delay.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

Thus we have proposed the Connectivity Equivalence Partitioning Algorithm and implemented in the ns network simulator. This algorithm partitions the mobile ad hoc network into equivalence partitions in which the nodes are equivalent in a routing perspective. First, we divide the nodes in to partitions and elect an active node for each partition. All the other nodes in the partition go to the sleep state and conserve power. Though there is a decrease in the connectivity of the network, greater amount of power is saved. The simulation results have shown that CEP gives about 40 to 50 % decrease in power consumption. The delay and packet delivery ratio are not affected significantly. As power is a very important metric in Mobile Ad Hoc networks, we can tradeoff connectivity with it.

5.2 FUTURE WORK

1. Different heuristics related to the rotation of the active nodes are to be analyzed so that all the nodes in the network are treated evenly and the overall network lifetime increases.
2. More evaluation of the partitioning algorithm should be performed, to reduce its convergence time and the adaptability to network mobility.
3. The cases in which the active node moves far from the remaining nodes and the value of the optimal time after which the partitioning algorithm must be undertaken should be analyzed.
4. The time for which a node must be awake as the active node in a partition and the time interval after which the equivalence partitioning algorithm must be reiterated must be determined.

5. The cases in which how a node in the off state can handle packets originated from it of destined to it must be analyzed. We propose to combine the topology maintenance algorithm with 802.11 power saving mode algorithm to overcome this challenge.

REFERENCES

1. C. Perkins. Ad hoc on demand distance vector (AODV) routing. *Internet-Draft, draft-ietf-manetaodv-04.txt*, pages 3-12, October 1999.

2. J. Broch, D. B. Johnson, and D. A. Maltz. The dynamic source routing protocol for mobile ad hoc networks. *INTERNET-DRAFT, draft-ietf-manetdsr- 03.txt,*, October 1999.

3. C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of the ACM SIGCOMM*, pages 234–244, August 1994.

4. V Park and S Corson, "Temporally ordered routing algorithm Version 1 fundamental specification *INTERNET-DRAFT, draft-ietf-manet-tora-02.txt,*, October 1999.

5. J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva. A performance comparison of multihop wireless ad hoc network routing protocols. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pages 85–97, October, 1998.

6. P. Johnsonn, T Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario Based performance analysis of routing protocols for mobile ad-hoc networks. In the proceedings of ACM/IEEE International Conference on Mobile Computing and Networking page 174-185, August 1999

7. S. R. Das, C. E. Perkins, and E. M. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proceedings of the IEEE Infocom*, pages 3–12, Tel Aviv, Israel, March 2000.

8. M Stemm and R H Katz. Measuring and reducing energy consumption of network interfaces in hand held devices. IEEE Transactions on Communications. August 1997.

9. O. Kasten. Energy Consumption. ETH – Zurich, Swiss Federal Institute of technology. Available at http:// www.inf.schx.ch/~kasten/research/bathtub/energy_consumption.html, 2001

10. B. Chen, K. Jamieson, H. Balakrishnan, R. Morris SPAN. *In the proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom) (Rome Italy, July 2001)*

11. G J Pottie and W J Kaiser. Embedding the internet: wireless integrated network sensors. Communication of the ACM May 2000

12. S Singh, C Ragavendra PAMAS: Power aware multi-access protocol with signaling for ad hoc networks. ACM Computer Communication Review. July 1998

13. Xu, Y., Heidemann, J., Estrin, D. Geography informed Energy conservation for AdHoc Routing. *In the Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking(MobiCom)(Rome Italy, July 2001)pp.70-84.*

14. The CMU Monarch Project: The CMU monarch project wireless and mobility extension to ns. http://www.monarch.cmu.edu

15. Anand Prabhu Subramanian, Connectivity Based Equivalence Partitioning of Nodes to Conserve Energy in Mobile Ad Hoc Networks, *In the proceedings of the Fifth International Conference on Ubiquitous Computing (Washington Seattle, October 2003)*

APPENDIX I

INSTALLATION INSTRUCTIONS

In order to install ns-2 network simulator unzip the ns-allinone-2.1b8a.gzip file to your home directory and execute the ./install file in the ns-allinone-2.1b8a directory.

After successful installation the installation file would look like.

tcl8.3.2: ~/ns-allinone-2.1b8a/{bin,include,lib}
tk8.3.2: ~/ns-allinone-2.1b8a/{bin,include,lib}
otcl: ~/ns-allinone-2.1b8a/otcl-1.0a7
tclcl: ~/ns-allinone-2.1b8a/tclcl-1.0b11
ns: ~/ns-allinone-2.1b8a/ns-2.1b8a/ns
nam: ~/ns-allinone-2.1b8a/nam-1.0a10/nam
xgraph: ~/ns-allinone-2.1b8a/xgraph-12.1
gt-itm: ~/ns-allinone-2.1b8a/itm, edriver, sgb2alt, sgb2ns, sgb2comns, sgb2hierns

Please put /home/anand/ns-allinone-2.1b8a/bin:/home/anand/ns-allinone-2.1b8a/tcl8.3.2/unix:/home/anand/ns-allinone-2.1b8a/tk8.3.2/unix into your PATH environment; so that you'll be able to run itm/tclsh/wish/xgraph.

IMPORTANT NOTE

(1) You MUST put ~/ns-allinone-2.1b8a/otcl-1.0a7, ~/ns-allinone-2.1b8a/lib,
into your LD_LIBRARY_PATH environment variable.

If it complains about X libraries, add path to your X libraries
into LD_LIBRARY_PATH.

If you are using csh, you can set it like:

```
setenv LD_LIBRARY_PATH <paths>
```

If you are using sh, you can set it like:

```
export LD_LIBRARY_PATH=<paths>
```

(2) You MUST put ~/ns-allinone-2.1b8a/tcl8.3.2/library into your TCL_LIBRARY environmental variable. Otherwise ns/nam will complain during startup.

(3) [OPTIONAL] To save disk space, you can now delete directories tcl8.3.2
and tk8.3.2. They are now installed under /1b8a/{bin,include,lib}

After these steps, you can now run the ns valid

```
cd ns-2.1b8a; ./validate
```

APPENDIX II

SIMULATION PARAMETERS

Channel Type	Channel/WirelessChannel
Radio Propagation Model	ropagation/TwoRayGround
Network Interface Type	Phy/WirelessPhy
MAC Type	Mac/802_11
Interface Queue Type	Queue/DropTail/PriQueue
Link Layer Type	LL
Antenna	Antenna/OmniAntenna

X dimension of the topography	1500
Y dimension of the topography	300
Traffic Pattern	"../mobility/scene/cbr-60-1-15-10"
Scenario	"../mobility/scene/scen-800x300-20-2-20-100"

Maximum Packet in IFQ	50
Number of Nodes	60
Seed for random generator	0.0
Simulation Time	400
Ad hoc Routing protocol	AODV
Initial Energy	500
Infinite Energy	20000

```
# Other default settings

LL set mindelay_                50us
LL set delay_                   25us
LL set bandwidth_               0

Agent/Null set sport_           0
Agent/Null set dport_           0

Agent/CBR set sport_            0
Agent/CBR set dport_            0

Agent/TCPSink set sport_        0
Agent/TCPSink set dport_        0

Agent/TCP set sport_            0
Agent/TCP set dport_            0
Agent/TCP set packetSize_       1460

Queue/DropTail/PriQueue set Prefer_Routing_Protocols    1

# unity gain, omni-directional antennas
# set up the antennas to be centered in the node and 1.5 meters above it
Antenna/OmniAntenna set X_      0
Antenna/OmniAntenna set Y_      0
Antenna/OmniAntenna set Z_      1.5
Antenna/OmniAntenna set Gt_     1.0
Antenna/OmniAntenna set Gr_     1.0
```

```
# Initialize the SharedMedia interface with parameters to make
# it work like the 914MHz Lucent WaveLAN DSSS radio interface
Phy/WirelessPhy set CPThresh_  10.0
Phy/WirelessPhy set CSThresh_  1.559e-11
Phy/WirelessPhy set RXThresh_  3.652e-10
Phy/WirelessPhy set Rb_        2*1e6
Phy/WirelessPhy set Pt_        0.28183815
Phy/WirelessPhy set freq_      914e+6
Phy/WirelessPhy set L_         1.0
```