CCIT Report #552
August 2005

Threshold-Related Throughput – A New Criterion for Evaluation of Sensor Network Performance*

Ilya Ledvich, Adrian Segall
Department of Electrical Engineering
Technion – Israel Institute of Technology
Haifa 32000, Israel
iledvich@tx.technion.ac.il, segall@ee.technion.ac.il

Abstract

Energy efficient and power aware protocols are of utmost importance in Sensor Networks. The most popular criteria, so far, for evaluating performance of energy-aware protocols are *lifetime* and *throughput*. One of the main contributions of the present report is to show that those criteria are often insufficient indications of the algorithm performance. Here we propose a new criterion, named *threshold-related throughput*, which provides a much better measure of the algorithm performance. The other main contribution is an extensive investigation of a large variety of routing protocols and routing cost metrics activated on a variety of Sensor Networks topologies and initial energy configurations. Performance of these protocols and configurations is studied and compared using the new criterion.

1 Introduction

1.1 Overview and Related Works

The research interest in different aspects related to deploying and further exploitation of Sensor Networks has been increasing in the few last years. A Sensor Network can be quickly and easily deployed and thus is suitable and very attractive for many environmental, commercial and military

* This research was supported by The Israel Science Foundation , Grant No. 148/03

applications. A general-purpose sensor network is commonly a dense network that consists of a large number of energy-constrained nodes; it is likely to be deployed in difficult access regions and to be remotely operated by only a few operators. One can conclude therefore that energy becomes the most critical resource. As a result, conserving energy should be a primary demand made of the protocols designed for such networks. This report addresses an important problem related to Sensor Network management – the problem of *online* message routing in a general-purpose sensor network.

Next, we briefly review previous related works. Heuristic energy aware routing algorithms whose routing objective is to maximize *network lifetime* can be found in [1], [2], [4], [6], [8], [10]. The routing metrics suggested in the above works try to maximize network lifetime by maximizing minimal residual energy [2], minimal residual link capacity [1], battery efficiency [10], or by minimizing total and maximal battery cost [4]. Works [3], [5] address the *online* routing problem where neither the sequence of future generated packets nor the originated packet rates are known in advance. The routing objective of the algorithm proposed in [3] is to maximize the total number of messages sent over the network (*network capacity*). The objective of the algorithm proposed in [5] is to maximize *network lifetime*. Performance of routing algorithms was widely studied and evaluated by means of simulations.

In this report, we present the results of an extensive investigation we have performed of a large variety of routing protocols. We have studied a variety of schemes and methods for evaluating and comparing the protocols. We show that the most popular criteria - *lifetime* and *throughput* - currently used in the literature, are often not sufficient for good evaluation of algorithm performance. We introduce a new method, named *Threshold-Related Throughput,* that provides a much more reliable indication of the algorithm performance.

1.2 Network and Energy Consumption Models

The Sensor network can be modeled as a directed graph $G(V,L)$, where V is the set of nodes and L is the set of directional links. The set of the nodes V may consist of several subsets. One subset is the subset of data sources and sinks, nodes that transmit and receive information respectively. In networks with a hierarchical structure, there may be a subset of cluster-head nodes, which gather and relay information, and may or may not generate data. In sensor networks with connectivity to the outside world, there may exist a subset of base stations, which provide connectivity between the sensor network and the outside world. Each unit, depending on the roles it plays, can be a member of one or more subsets.

Every device j in the network, with exception of the base stations, has a finite *initial energy* E_j accumulated in its battery. We associate this parameter to each node j within the graph $G(V,L)$. This implies that, unlike most kinds of graphs, in our network, there are weighted nodes in addition to weighted links. The *residual energy* of node j at time t is denoted by $E_j(t)$. As stated above, a node in a sensor network may be engaged in different kinds of activity, and thus its energy is consumed by several modules, like sensor and signal processing devices, computation and radio units, etc. The radio unit in transmit mode is considered to be the main consumer of energy resources. Therefore, we neglect the energy consumed in other modes, like idle, sleep, sense, and by other modules and concentrate on energy dissipation when a node transmits and receives packets.

The amount of energy consumed by some node j for transmitting a packet directly to another node k is denoted by e_{jk}^{tx}. This parameter is called *packet transmission energy cost* of the link (j,k). We associate it with each link from the set L. The energy consumed by a receiving node is denoted by e_k^{rx} and is referred to as the *receiving energy cost*. Generally, the transmission energy e_{jk}^{tx} is a function of a packet's length and the distance between j and k, while the receiving energy e_k^{rx} does not depend on the distance between j and k. Our assumption is that all data packets in the system have the same length, incorporated in the calculation of e_{jk}^{tx}.

When a data packet travels via on a multi-hop path, it is received by each hop for further transmission to the next one. The *energy cost of forwarding a packet* e_{jk} is the sum of the energies consumed by each node participating in this process. The energy consumed by each node, is equal to the sum of receiving and transmitting energy costs.

We say, that node j is *connected* to node k at time t by a directional link (j,k), if the residual energy in j at this time, denoted by $E_j(t)$, is equal to or greater than the packet transmission cost e_{jk}^{tx} and the residual energy in k is equal to or greater than the packet receiving cost e_k^{rx}. In other words, node j is connected to node k if at least one packet can be transmitted by j and directly received by k.

We assume that once the battery charge has been depleted, it cannot be replenished. When its energy is exhausted, the node becomes *isolated* or *dead*. Such a node cannot forward any packets whether originated by it or received from other nodes and therefore cannot participate in the routing process. Although the amount of residual energy in such an isolated node may allow it to still receive packets, we want to avoid the situation when packets are sent to isolated nodes. There is no way for other nodes in a network to distinguish between a *dead* node with a completely exhausted battery and an *isolated* node with some positive battery charge, because no interaction with such a node is possible. For this reason, we remove such nodes from the set of nodes V as soon as they become isolated. All incoming links of a node that becomes isolated are removed from the set of links L as well. Thus, the set of nodes V at any time t contains active nodes only, and the set of the links L consists only of active links.

We can define the set of links L at time t as follows:

$$L(t) = \{(j,k): j,k \in V(t), j \neq k, e_{jk}^{tx} \leq E_j(t), e_k^{rx} \leq E_k(t)\}. \qquad (1.1)$$

Node k is said to be a *neighbor* of the node j if link $(j,k) \in L(t)$. In other words, all nodes with which the node j has direct links are called j's neighbors. Let $N_j(t)$ denotes the *set of neighboring nodes* or *neighborhood* of the node j at time t. We can define it as follows:

$$N(t) = \{k \in V(t) : (j,k) \in L(t)\}. \qquad (1.2)$$

Node j is declared *active* if its neighborhood is a nonempty set, it is called *isolated* otherwise. See example in Figure 1.1.

The *path* or *route*, denoted by p or P, connecting source node S with destination node D is defined as a sequence of nodes (hops), where the first and the last elements are S and D respectively. Alternatively, the path can be defined as a sequence of links, where the first element is some outgoing link of node S and the last element is some incoming link of node D.

The next step is to define the cost metric function for each link in the graph and the method used for calculation of a multi-hop path cost. Let $C_{jk}(t)$ be a cost or weight of a link (j,k) at time t and $C_P(t)$ be a cost of a path P at time t. The selection of a proper cost function is one of the greatest challenges in developing routing algorithms (see Fig.1.2). In this work, we focus on *on-line* routing problems, namely situations when, neither the sequence of future packets nor the generation rates of packets are known in advance.

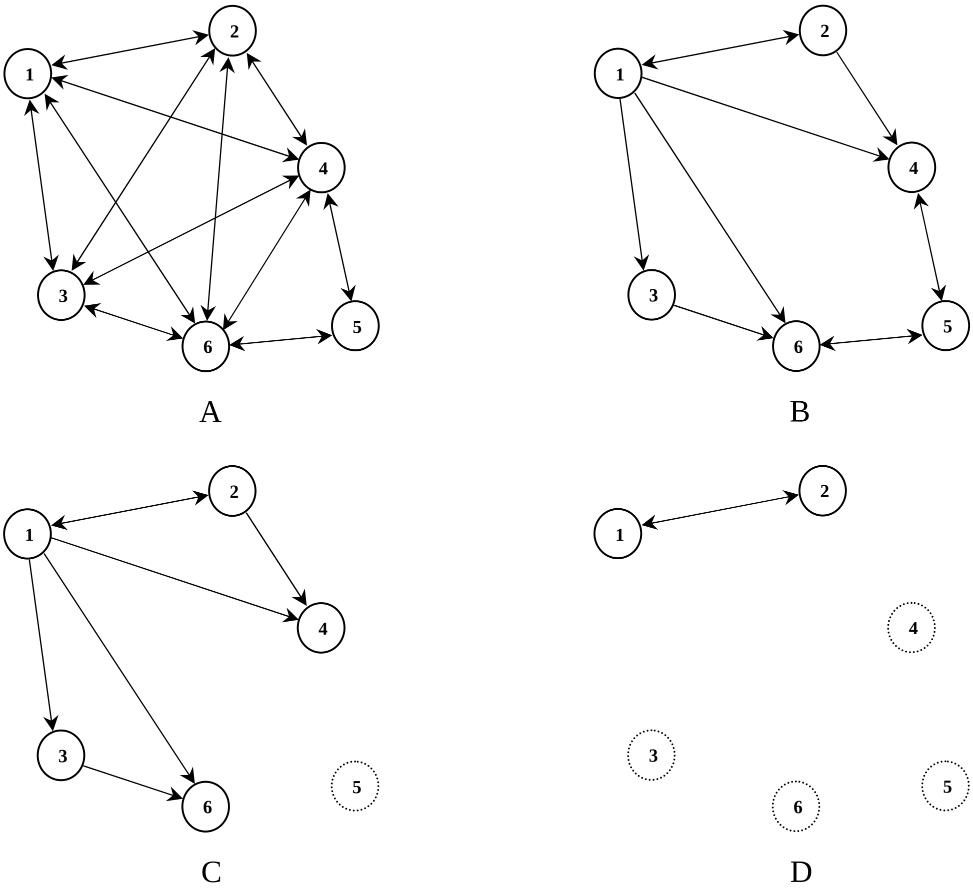


Figure 1.1. We start with a strongly connected graph (see Figure 1.1.A). After a while, it looks as in Figure 1.1.B. When node 5 finally exhausts its battery, it becomes *isolated* and is removed from the graph (Figure 1.1.C). So do nodes 4 and 6 which now are not connected to any *active* node. This fact causes node 3 to become *isolated* as well, because its only *neighbor* 6 is removed from the set of the graph's nodes. Nodes 1 and 2 are still active because they are connected to each other (Figure 1.1.D).

We now discuss several approaches to modeling the energy consumption process. As stated above, nodes in a Sensor Network may be engaged in different kinds of activity: gathering information about some environmental phenomenon, processing analog and digital signals, maintaining connectivity with its closest neighborhood, forwarding routing and control packets, etc. In the present work, we focus on the message routing problem, thus we assume that all other node activities are less significant and therefore the related energy dissipation can be neglected.

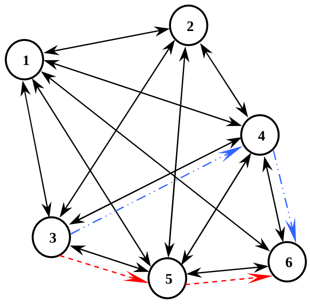


Figure 1.2. Assume graph $G(V,L)$ as in Figure 1.2. Let us define link cost function as follows: $C_{jk}(t)=1, \forall (j,k)\in L(t), \forall t$. In other words, each link has a unit cost as long as the link exists. In addition, we define the calculation method as the arithmetical sum and the comparison rule as the less the better. Minimum hop routes are optimal. Thus, the shortest path from node 3 to node 6 is either $P_1=\{3,5,6\}$ or $P_2=\{3,4,6\}$. One can see that $C_{P_1}(t)=C_{P_2}(t), \forall t$. It is clear, that such a routing scheme does not take into account energy resources in the network.

Using the model of exponential decay of radio signal, the energy cost of transmitting a packet over link (j,k) can be calculated as follows:

$$e_{jk}^{tx}=(\alpha\cdot d_{jk}^{n}+\beta)\cdot l, \tag{1.3}$$

where l represents the length of a packet in bits, α and β are some real numbers, d_{jk} is the physical distance between nodes j and k, and n is a real number, named *loss index*, that typically takes on a value between 2 and 4. The constant α denotes the energy dissipated in the transmitter amplifier per bit transmitted over unit of distance; while β is a distance-independent term representing the energy dissipated in the transmitter electronics per bit ([1] – [10]); sometimes β is called overhead energy [13]. In addition, the energy consumed at the receiving node k for reception a packet of l bits is distance-independent and can be modeled as follows:

$$e_{k}^{rx}=\gamma\cdot l, \tag{1.4}$$

where γ is some real number denoting the energy consumed in receiver electronics per received bit. Using this model, it is easy to compute the optimal number and location of relay nodes, which can be placed between source node j and destination k in order to minimize the total energy dissipated on transferring a unit of information from j to k. As it was shown in [11], the optimal number can be computed as follows:

$$N_{opt} = \left\lfloor \frac{d}{d_{char}} - 1 \right\rfloor \text{ or } \left\lceil \frac{d}{d_{char}} - 1 \right\rceil, \tag{1.5}$$

where d is the distance between j and k, and d_{char}, called the characteristic distance, is completely determined by α, β, γ and n:

$$d_{char} = \sqrt[n]{\frac{\beta + \gamma}{\alpha(n-1)}} . \tag{1.6}$$

The values of the parameters α, β, γ and n, widely vary in the literature. According to the so-called first order radio model used in [7], [8], [9], typical values of the above-mentioned parameters are $\beta = \gamma = 50$ nJ/bit, $\alpha = 100$ pJ/bit/m^2, and $n = 2$. Using these values one can conclude that the distance-independent term in equation 1.3 outweighs the distance-dependent one as long as the transmission distance does not exceed 22 meters; furthermore, for distances shorter than roughly 70 meters its contribution to total energy dissipation is still significant and therefore cannot be neglected. In addition, the energy consumed in the receiving node has a quite significant value as well.

A simpler model, used in [12], [13], assumes that the energy consumed in the receiving node is less significant than indicated above. Typical values of other parameters as they appear in [12] are: $\beta = 50$ nJ/bit and $\alpha = 0.0013$ pJ/bit/m^4 for loss index $n = 4$. One can see that if the distance between receiver and transmitter is shorter than approximately 80 meters, the contribution of the overhead energy β in equation (1.3) is more significant than that of the distance-dependent term $\alpha \cdot d^n$. In [13] it was shown how performance of routing algorithms can be affected if the overhead energy is neglected during computation of transmitting cost of the graph's links.

Finally, the simplest model of consumption of energy is used in [1], [2], [3], [5], [10]. According to it, the values of β and γ are small in comparison with $\alpha \cdot d^n$ and therefore the energy consumed by both receiver and transmitter electronics can be neglected. In addition, some minimal amount of energy e^{tx}_{min}, reflecting the sensitivity of the receiver's equipment, has to be introduced. Finally, the energy cost of transmitting a packet of l bits over some link *(j,k)* can be computed as follows:

$$e^{tx}_{jk} = max(e^{tx}_{min}, \alpha \cdot d^n_{jk}) \cdot l . \tag{1.7}$$

In order to minimize the total transmitting energy spent on transferring a packet from source node S to destination D, the optimal number N_{opt} of relaying nodes that should be placed between S and D is:

$$N_{opt} = \left\lfloor \frac{d}{d_{min}} - 1 \right\rfloor \text{ or } \left\lceil \frac{d}{d_{min}} - 1 \right\rceil . \tag{1.8}$$

Specific values of parameters α, e^{tx}_{min} and n found in the literature widely vary.

Now we shall discuss several approaches for determining the ability of Sensor nodes to adjust the transmitted signal energy, in order to reach the desired destination. The simplest way is to assume that the transmission power is not tunable, and thus the transmission range is fixed. Such a model has a few indisputable advantages. Firstly, such device should not be too complicated and thus not too expensive. Secondly, establishing and maintaining network connectivity appears to be a relatively simple procedure. On the other hand, when the distance between two nodes j and k is greater than their communication range, no interconnection between j and k is possible, even if the energy resources in the nodes allow that. Such drawback can be eliminated if nodes posses tunable transmitters. It is clear that the maximal transmission range of such nodes is always limited.

2 Energy-aware routing algorithms

In this part of the report, we shall discuss several energy-aware routing algorithms. The algorithms can be divided into four groups, according to the methods and rules they use for calculating and comparing routing paths: *minimum total cost routing*, *min-max cost routing*, *max-min cost routing* and *hybrid cost routing*. Within each group, algorithms differ by the link cost function. The groups differ by the scheme for computing the cost of the entire path and by the method for comparing the quality of the path. The summary of the routing schemes, divided onto the four groups, can be found in Table 2.1.

Consider a Sensor network modeled by graph $G(V,L)$. The routing algorithm provides the best route from a source node j to a destination node k according to some criterion if such a route can be found or rejects the packet if no feasible route exists. Any *minimum cost routing* algorithm minimizes the total (aggregated) cost of forwarding the packet along the entire route. In other words, among all possible routes from source node S to destination node D, the minimum cost routing algorithm selects the one with the minimum total cost. The total cost is calculated as the sum of the link costs along the route. The pseudo-code of a generic minimum cost routing algorithm that utilizes some cost function $C_{jk}(t)$ can be found in Appendix 2.C.

In contrast to the algorithms from the previous group, *min-max cost routing* algorithms are oriented to minimize the maximal cost of links in the path rather than the total path cost. The selected route is the one whose maximal link cost is minimal.

The third group of algorithms is *max-min cost routing*. These algorithms route packets along paths with maximum minimal link cost.

The *hybrid (mixed) cost routing* is some combination of the three previous routing techniques. Each algorithm in the group applies its own methods and rules. The advantages and disadvantages of the various algorithms are discussed in the next three sub-sections.

2.1 Minimum Cost Routing Metrics

Consider a path P connecting source node S with node D and containing m additional intermediate nodes:

$$P = \{(S, R_1),(R_1, R_2),\ldots,(R_m, D)\} = \{(R_0, R_1),(R_1, R_2),\ldots,(R_m, R_{m+1})\}, \tag{2.1}$$

where R_0 and R_{m+1} denote source S and destination D respectively and R_j denotes the j-th intermediate node (hop) in the path P. Let $C_{R_j,R_{j+1}}(t)$ be the cost of the link $(R_j, R_{j+1}) \in P$. The cost of the route P is computed as follows:

$$C_P(t) = \sum_{j=0}^{m} C_{R_j,R_{j+1}}(t). \tag{2.2}$$

The simplest and most intuitive algorithm for the energy-aware routing problem is to minimize the total cost of forwarding a packet. Intuitively, it seems to be very energy efficient. The first algorithm, referred to as MTER (Minimum Transmission Energy Routing), in Table 2.1 exploits this simplest cost function – the forwarding energy cost:

$$C_{jk}(t) = e_{jk}. \tag{2.3}$$

According to this criterion, the cost of the path is equal to the sum of the energies consumed in each hop. The MTER path is the path with minimal total energy consumed by forwarding a packet along it. However, there is a significant drawback in this approach. The algorithm uses the same route each time it sends packets from a given source to a given destination. Thus, the nodes on the MTER path will quickly deplete their batteries. Since the algorithm does not take into account residual node energy, it might quickly lead to network split into several disconnected sub networks. See example in the Figure 2.1.

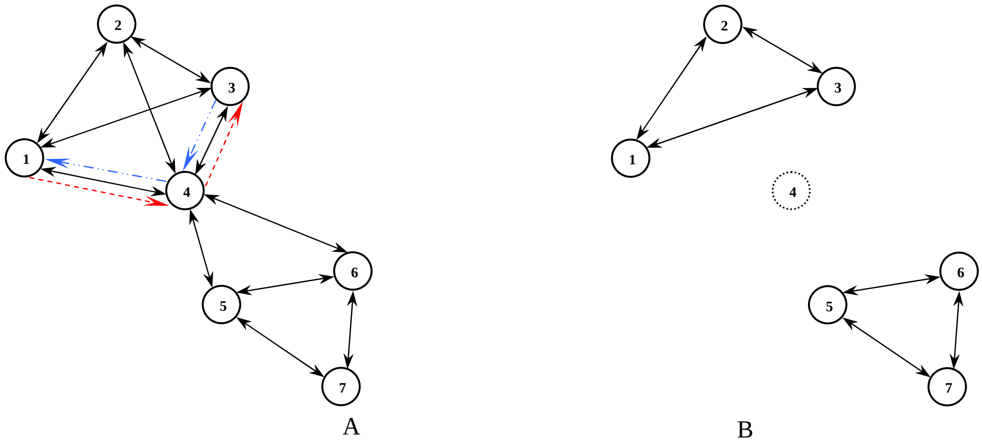


Figure 2.1. The MTER routing scheme is applied in the graph of Fig 2.1.A. The cost function used by MTER is the transmission energy. Assume that the total cost of path $\{1,2,3\}$ is only ε larger than the cost of the path $\{1,4,3\}$, so these two routes are almost equivalent. However, according to the MTER routing policy, node 1 always selects its neighbor 4 in order to forward packets to 3, as long as this path exists. Eventually node 4 will be unable to forward any more packets because of lack of energy. At that instant, the strongly connected network is split onto two disconnected parts (Fig.3.B). The lifetime of node 4 and therefore, the connectivity time of the network could be prolonged if transmission were balanced between two alternative almost equivalent routes, $\{1,4,3\}$ and $\{1,2,3\}$.

The drawbacks of the previous routing scheme can be eliminated by using a cost function that depends on residual energy. The cost function used by the MBCR (Minimum Battery Cost Routing) algorithm is inversely proportional to the residual energy in the node and does not depend on transmission energy:

$$C_{jk}(t) = 1/E_{jk}(t)\,. \tag{2.4}$$

Therefore, all outgoing links of some node j at any time instant t have the same weight. When the node participates in forwarding packets and therefore consumes some amount of energy, the costs of its outgoing links increase. This algorithm, in contrast to the previous one, can select different routes for the same source-destination pair packets each time and therefore balances the forwarding load among several paths. Moreover, the MBCR algorithm tends to include into the route nodes with a large amount of residual energy and to avoid using the energy-weak nodes. On the other hand, the route selected by MBCR might be very energy-wasteful, since the algorithm does not take into account the total transmission cost of forwarding the packet. See example in Figure 2.2.

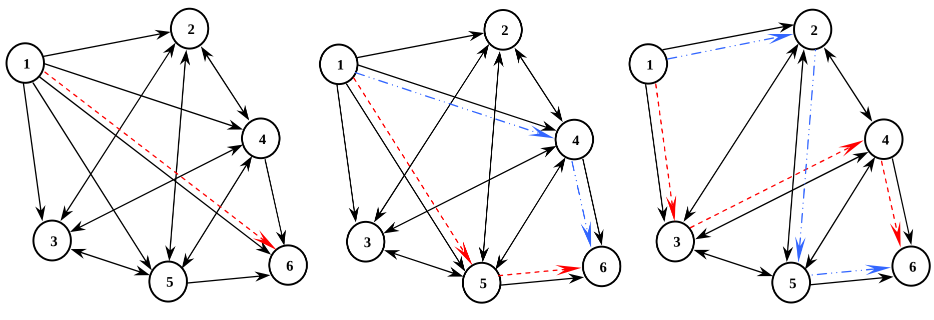


Figure 2.2. Assume graph $G(V,L)$ as in Figure 2.2.A. For the sake of simplicity, assume that there is a single source (node 1) and a single destination (node 6) in the network. In addition, assume that the two following conditions are satisfied:

$$e_{16} > e_{14} = e_{15} > e_{23} = e_{25} = e_{34} > e_{12} = e_{13} = e_{45} > e_{46} = e_{56} > e_{35} = e_{24},$$

$$e_{\{1,6\}} > e_{\{1,5,6\}} = e_{\{1,4,6\}} > e_{\{1,3,4,6\}} = e_{\{1,2,5,6\}} > e_{\{1,3,5,6\}} = e_{\{1,2,4,6\}},$$

where $e_{\{i,j,\dots,k\}}$ means energy cost of the path $\{i, j,\dots,k\}$. Routes are selected according to the MBCR routing policy. Firstly, the source sends packets directly to the destination as long as the battery charge allows this (Fig.2.2.A). After that, the source alternates the two-hop paths $\{1,5,6\}$ and $\{1,4,6\}$ (Fig.2.2.B) until these routes become infeasible. At the next stage the source finds other paths in order to send packets, say two three-hop routes ($\{1,3,4,6\}$ and $\{1,2,5,6\}$) are taken in turn by the source (Fig.2.2.C). The routes are taken in this order because the following condition is always satisfied:

$$C_{\{1,6\}}(t) < C_{\{1,5,6\}}(t), C_{\{1,4,6\}}(t) < C_{\{1,3,4,6\}}(t), C_{\{1,2,5,6\}}(t), C_{\{1,3,5,6\}}(t), C_{\{1,2,4,6\}}(t), \forall t,$$

where $C_{\{i,j,\dots,k\}}(t)$ is the MBCR cost of path $\{i, j,\dots,k\}$ at the time t. One can conclude that using either $\{1,3,4,6\}$ or $\{1,2,5,6\}$ paths (as it would be done by the MTER algorithm) can yield better performances (in terms of *lifetime* or *total throughput*) because it allows more efficient utilization of the energy resources of node 1.

Note that the drawback stated above is not relevant in systems with nodes which can transmit only at some fixed energy level, the same level for all devices. In that case the MBCR metric is the same as the MREPcapsum metric below.

As shown before, cost functions that use only the transmission cost or only the residual energy, do not provide good measures for the solution of the energy efficient routing problem. The next block of algorithms introduces metrics, which include information about both parameters.

The next algorithm, MREPcapsum ("MREP" is an abbreviation for Minimum Residual Energy Path, "cap" means "capacity", "sum" means that the total cost of path is computed as a sum of link costs composing it) combines the two previous schemes and uses a metric, which is proportional to the transmission energy divided by the residual energy.

$$C_{jk}(t) = e_{jk} / E_{jk}(t) \,. \tag{2.5}$$

This cost function is in fact inversely proportional to the *residual link capacity* – the number of packets that can still be sent on this link, if the origin node of this link would transmit packets over this link only. This algorithm tries to abstain from forwarding messages via energy-expensive links and energy-exhausted nodes, making the distribution of the residual nodes energy of the nodes more balanced. An example appears in Figure 2.3.

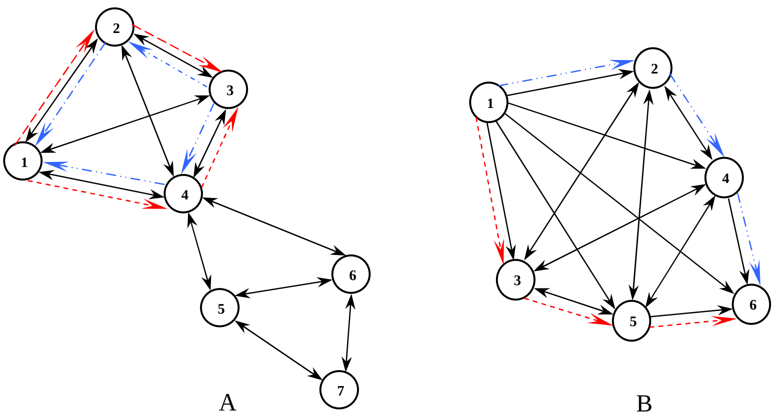


Figure 2.3. It can easily be seen that using MREPcapsum criterion in the two previous examples can yield better results. The total connectivity time in the first case is prolonged because the transmission load bestowed on the critical node *4* is reduced by using an alternate path (Fig. 2.3.A). The total throughput delivered to the destination in the second case is increased because this algorithm prevents sending packets over too exhausted routes (Fig. 2.3.B).

The CMAX (Capacity MAXimization) algorithm uses a more complex cost function, which is linear in the transmission energy e_{jk} and is exponential in a parameter $a_j(t)$, named energy utilization:

$$C_{jk}(t) = e_{jk} \cdot (\lambda^{a_j(t)} - 1), \tag{2.6}$$

where $a_j(t)$ is the percentage of the initial energy that has already been spent at the node:

$$a_j(t) = 1 - E_j(t)/E_j. \tag{2.7}$$

The specific value of the parameter λ is not so critical as long as λ is sufficiently large [3]. Let us now discuss the possible problems with this criterion. First, one can easily see that if some node j has never participated in forwarding packets, the weights of its outgoing links are equal to zero. Second, the cost function depends on utilized energy, but not on the residual one. It is clear, that because of these reasons, packets might be transmitted along very inefficient paths.

We believe that the performance of the CMAX algorithm can be significantly improved if the cost function is modified as follows:

$$C_{jk}(t) = e_{jk} \cdot \lambda^{a_j(t)}. \tag{2.8}$$

One can see that with this metrics, the initial cost of each link is now the transmission energy associated with it.

2.2 Min-Max and Max-Min Cost Routing Metrics

We now examine the next group – the *min-max cost algorithms*. As said before, the cost of a min-max path is equal to the cost of the link with the largest cost. Consider a path P connecting source node S with node D and containing m additional intermediate nodes (see Equation 2.1).

When some min-max cost routing scheme is used, the cost of the route P is computed as follows:

$$C_P(t) = \max_{\forall (R_j, R_{j+1}) \in P} \left(C_{R_j, R_{j+1}}(t) \right). \tag{2.9}$$

Conventional shortest path algorithms cannot be applied in order to find the min-max route. A few possible solutions for this problem were proposed in [1], for instance using a slightly modified Bellman-Ford algorithm. The conventional equation for distance update at next step:

$$D_j^{(n+1)} = \min \left[\min_{i \in V} \left\{ C_{ji}(t) + D_i^{(n)} \right\}, D_j^{(n)} \right]. \tag{2.10}$$

that can be used by min-cost algorithms, is modified as follows:

$$D_j^{(n+1)} = min\left[\min_{i \in V} \left\{ max\left(C_{ji}(t), D_i^{(n)} \right) \right\}, D_j^{(n)} \right]. \quad (2.11)$$

where $D_j^{(n)}$ is the cost of a path to a given destination, provided by some node j at n-th step of Bellman-Ford algorithm. The equation $max\left(C_{ji}(t), D_i^{(n)} \right)$ means the cost of the path from j to destination provided it passes neighbor k. At the next iteration step, node j makes a decision and either prefers its neighbor k, which provides the shortest min-max path among all j's neighbors, or stays with its previous path. The regular Bellman-Ford algorithm and its min-max version can be found in Appendix 2.B.

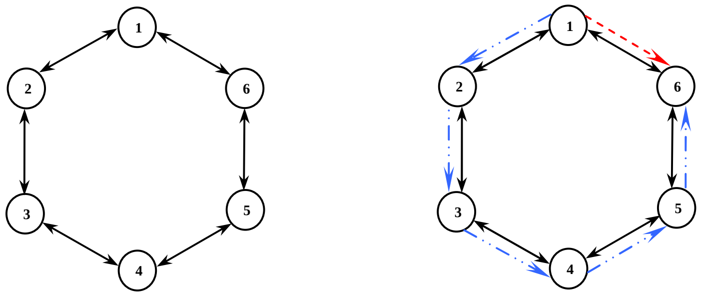


Figure 2.4. Assume graph $G(V,L)$ as in Fig.2.4.A. In addition assume that all links in the graph have the same weights: $C_{jk} = C, \forall (j,k) \in L$. Consider paths $\{1,6\}$ and $\{1,2,3,4,5,6\}$ (see Fig.2.4.B). The minimum cost of the first path is five times smaller than that of the second path. On the other hand, both routes have the same min-max cost, and therefore each of them can be selected by some careless min-max shortest path algorithm.

The main problem of all min-max algorithms is their sensitivity to the implementation of the shortest path algorithm responsible for selecting a proper route. The min-max cost of a path is equal to the maximal value among the links composing it. Therefore, if the cost of some path P is C_P, then adding a number of less expensive links to the path does not increase its cost. This fact shows that very inefficient routes might be selected. An example of the described problem can be found on Figure 2.4.

Let us now briefly discuss the algorithms. The MMBCR (Min-Max Battery Cost Routing) algorithm uses the same metric as MBCR. Furthermore, a path selected by MMBCR might include redundant nodes, so the algorithm performance (in terms of throughput and first packet loss) might be even worse than that of its min-cost counterpart. An example can be found in Figure 2.5.

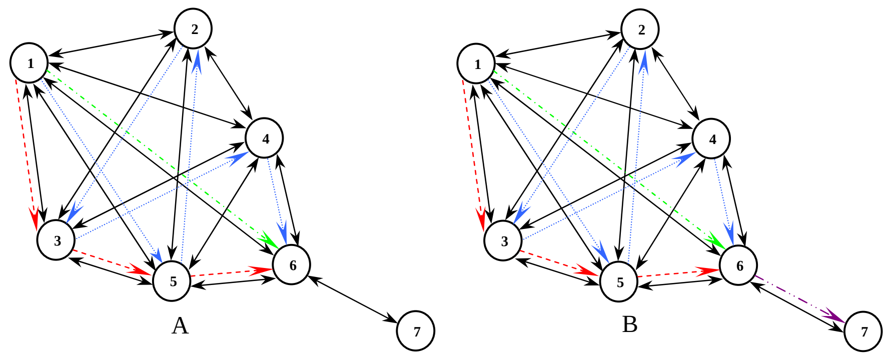


Figure 2.5. Assume graph $G(V,L)$ as in Fig.2.5.A. Let $C_{jk}(t)$ be the MMBCR cost of link (j,k), and $W_{jk}(t)$ be its MREP cost: $C_{jk}(t)=1/E_j(t), W_{jk}(t)=1/(E_j(t)-e_{jk})$. Assume that at some moment t', node 1 has the minimal value of residual energy among all nodes in the network: $E_1(t')\le E_j(t'), \forall j\in V$. Consider the following routes: the MTER path $\{1,3,5,6\}$, the direct path $\{1,6\}$ and the path $\{1,5,2,3,4,6\}$ (Fig.2.5.A) that contains several redundant nodes. Assume that the energy costs of these routes (energy consumed by forwarding a packet along each of them) satisfy the following inequality: $e_{\{1,3,5,6\}}<e_{\{1,6\}}<e_{\{1,5,2,3,4,6\}}$. However, their current MMBCR costs are equal:

$$C_{13}(t')=C_{16}(t')=C_{15}(t')=1/E_1(t') \qquad (*)$$

$$C_{\{1,3,5,6\}}(t')=C_{13}(t');\, C_{\{1,6\}}(t')=C_{16}(t');\, C_{\{1,5,2,3,4,6\}}(t')=C_{15}(t') \qquad (**)$$

$(*)\cup(**)\Rightarrow C_{\{1,3,5,6\}}(t')=C_{\{1,6\}}(t')=C_{\{1,5,2,3,4,6\}}(t')=1/E_j(t')$.

Thus at the current stage all these quite different paths can be selected. Now compute the MREP costs of these routes. It is clear that their MREP costs are in general different, since transmission energy is now taken into account. Therefore, the following statement is always true:

$e_{15}>e_{16}>e_{13}\Rightarrow W_{13}(t)<W_{15}(t)<W_{16}(t)\Rightarrow W_{\{1,3,5,6\}}(t)\ne W_{\{1,6\}}(t)\ne W_{\{1,5,2,3,4,6\}}(t)$.

So, unlike the last example, the packet destined to node 6 will never be sent over the redundant path, since the MTER path is always "shorter": $W_{\{1,3,5,6\}}(t)<W_{\{1,5,2,3,4,6\}}(t)$. Suppose now that at some moment t'' the link $(6,7)$ has the maximal MREP cost. Then, (Fig.2.5.B): $C_{67}(t'')\ge C_{jk}(t''), W_{67}(t'')\ge W_{jk}(t''), \forall (j,k)\in L$, thus, the MMBCR and MREP costs of every path which contains this link is equal to $C_{67}(t'')$ and $W_{67}(t'')$ respectively:

$$C_{\{1,3,5,6,7\}}(t'')=C_{\{1,6,7\}}(t'')=C_{\{1,5,2,3,4,6,7\}}(t'')=C_{67}(t'')d,$$

$$W_{\{1,3,5,6,7\}}(t'')=W_{\{1,6,7\}}(t'')=W_{\{1,5,2,3,4,6,7\}}(t'')=W_{67}(t'').$$

One can conclude that all min-max algorithms are very sensitive to implementation of the shortest path algorithm.

As stated above, the primary drawback of the previous metric, that all outgoing links of some node have the same weights. However, different links have different transmission costs associated with them, so this fact should be reflected in the links' costs. It was done in MREPmax (as it was mentioned above, "MREP" is the abbreviation for Minimum Residual Energy Path, while "max" means that the total cost of path equals to maximal cost of links composing it) algorithm. As we can see, the link cost is inversely proportional to residual energy at the node after sending packet over this link:

$$C_{jk} = 1/(E_j(t) - e_{jk}) . \qquad (2.12)$$

Unlike the previous criterion where all links of some node have equal weight, this one breaks that equality. However, this improvement does not completely eliminate the main drawback of min-max routing technique: a number of "redundant" nodes might be found in selected route as before. See example in Figure 2.5.

According to the *max-min* approach, the cost of the path is given by the link with minimal cost, i.e. the path *bottleneck*.

The cost function utilized by the MRPC (Maximum Residual Packet Capacity) algorithm is equivalent to the residual link capacity – the number of packets that could be sent on the link, if no other links outgoing from the given node are used.

$$C_{jk}(t) = E_{jk}(t)/e_{jk} . \qquad (2.13)$$

2.2 Hybrid Routing Metrics

Let us now briefly discuss the last group of algorithms. As stated above, no generic method can be defined for algorithms from this group. We have considered the max-min zP_{min} routing algorithm, as it was proposed in [5]:

1. Find the path with the least power consumption, P_{min}, by using the Dijkstra algorithm.
2. Find the path with the least power consumption in the graph.
   If the power consumption $> z \cdot P_{min}$ or no path is found,
   Then the previous shortest path is the solution, stop.
3. Find the minimal $u_{jk}(t)$ on that path, let it be u_{min}.
4. Find all the edges whose residual power fraction $u_{jk}(t) \le u_{min}$ and remove them from the graph.

Goto 1.

Where

$$u_{jk}(t) = (E_j(t) - e_{jk}) / E_j \,. \tag{2.14}$$

is the normalized residual energy, and z is a parameter of a system. As we can see, the algorithm described above utilizes two kinds of metric:

1. the MTER metric is used in order to compute the total cost of the path,
2. the residual power fraction metric is applied to reduce the size of the graph by discarding too expensive links.

The running time of max-min zP_{min} algorithm is $O(log\,|L|\cdot(|L|+|N|\cdot log\,|N|))$, where $O(|L|+|N|\cdot log\,|N|)$ is the running time of the Dijkstra algorithm, which is called at most $O(log\,|L|)$ times.

The CMMBCR (Conditional Min-Max Battery Capacity Routing) algorithm exploits another hybrid scheme. It selects the MTER path as long as there is at least one that satisfies the following condition: all nodes in it have residual energy higher than some threshold γ defined in advance. Otherwise, it selects routes by using the MMBCR scheme. In addition, this algorithm can apply pure MTER or pure MMBCR routing schemes by choosing γ equal to zero or to the initial energy level respectively.

Lastly, the CMRPC (Conditional Maximum Residual Packet Capacity) algorithm is the MRPC equivalent of the CMMBCR algorithm [6]. In other words, this algorithm starts with the MTER routing scheme and at certain stage changes it to the MRPC scheme.

Appendix 2.A – Table of Energy-Aware Routing Metrics

Group	##	Name	Link Cost Function $C_{jk}(t)$	References
Minimum total cost	(1)	MTER , Pmin, MTPR,	e_{jk}	[1],[3],[4],[7]
	(2)	MBCR	$1/E_j(t)$	[4]
	(3)	MREPcapsum	$e_{jk}/E_j(t)$	[1]
	(4)	MREPsum	$1/(E_j(t) - e_{jk})$	[1]
	(5)	CMAX	$e_{jk} \cdot (\lambda^{a_j(t)} - 1)$, $a_j(t) = 1 - E_j(t)/E_j$	[3]
Min-max cost	(6)	MMBCR	$1/E_j(t)$	[4]
	(7)	MREPmax	$1/(E_j(t) - e_{jk})$	[1],[2]
	(8)	MREPcapmax	$e_{jk}/E_j(t)$	[1]
Max-min cost	(9)	MRPC	$E_j(t)/e_{jk}$	[6]
Hybrid cost	(10)	Max-min z-P_{min}	e_{jk} and $(E_j(t) - e_{jk})/E_j$	[5]
	(11)	CMMBCR	e_{jk} or $1/E_j(t)$	[4]
	(12)	CMRPC	e_{jk} or $E_j(t)/e_{jk}$	[6]

Table 2.1. Energy-aware routing algorithms and their metrics.

Appendix 2.B – The Bellman-Ford Shortest Path Algorithm and its Min-Max Cost Version

The pseudo-code of the Bellman-Ford shortest path algorithm is given below. For a given graph $G(V,L)$, it finds the length of the shortest paths from each node in the graph to a given node s and the next hop in this path:

Each node $j \in V$ performs the following steps:

1. **For** each $j \in V$ do:
   - $D_j^{(0)} \leftarrow \infty$;
   - $V_j \leftarrow j$;
   - $D_s^{(0)} \leftarrow 0$;
2. $n \leftarrow 0$;
3. **Repeat**
   - $D_j^{(n+1)} \leftarrow min\left[\min_{i\in V}\left\{C_{ji}(t)+D_i^{(n)}\right\}, D_j^{(n)}\right]$;
   - **If** $D_j^{(n)} \le \min_{i\in V}\left\{C_{ji}(t)+D_i^{(n)}\right\}$
     - **Then** $V_j^{(n+1)} \leftarrow V_j^{(n)}$;
     - **Else** $V_j^{(n+1)} \leftarrow argmin\left[\min_{i\in V}\left\{C_{ji}(t)+D_i^{(n)}\right\}\right]$;
   - $n \leftarrow n+1$;
   - Advertise $D_j^{(n)}$ to each j's neighbor;

   Until $D_j^{(n+1)} = D_j^{(n)}, \forall j \in V$;
4. Stop; each D_j contains the length from j to s; V_j is the next hop in the path.

Note that V and L are the set of the nodes and set of the links respectively; n means iteration number, $n \le |V|$; $C_{ji}(t)$ is the cost of link (j,i) at the time t; $D_j^{(n)}$ contains the length of the shortest path from j to s at step n.

The Bellman-Ford shortest path algorithm can be used by any minimum cost routing algorithm. When the min-max cost path is required, it must be modified as follows. The line

$$D_j^{(n+1)} = min\left[\min_{i\in V}\left\{max\left(C_{ji}(t), D_i^{(n)}\right)\right\}, D_j^{(n)}\right]$$

should replace the line

$$D_j^{(n+1)} = min\left[\min_{i\in V}\left\{C_{ji}(t)+D_i^{(n)}\right\}, D_j^{(n)}\right].$$

Appendix 2.C – Minimum and Min-Max Cost Routing Algorithms

The pseudo-code of a general minimum cost routing algorithm that utilizes some cost function C_{jk} can be found below. For the sake of simplicity, we consider a slotted time system. At the beginning of each time slot, some node s (the source) originates a packet, destined to another node d (the destination). If a route for this packet is found, the packet is delivered to its destination during the current time slot; otherwise, it is rejected by the system. At most one packet could be originated and delivered during a given time slot.

For a given packet originated by source node s at the beginning of time slot n and destined to node d in given network modeled by graph $G(V(n),L(n))$ perform the following steps:

1. **For** each link $(j,k) \in L(n)$ calculate its cost $C_{jk}(n)$.
2. Find the shortest path p from s to d in the graph by means of any shortest path algorithm (e.g. Dijkstra, Bellman-Ford etc.):
   - **If** the path is found: $p = \{s, j, k, \dots, m, d\}$
     - **Then** $P^{s \leftarrow d} = \{(s, j), (j,k), \dots, (m,d)\}$;
       - Forward the packet along the path p (forwarding of packet takes exactly one time slot)
     - **Else** $P^{s \leftarrow d} \leftarrow 0$;
       - Reject the packet.
3. $V(n+1) \leftarrow V(n)$;
   $L(n+1) \leftarrow L(n)$.
4. **For** each $(j,k) \in P^{s \leftarrow d}$ do:
   - $E_j(n+1) \leftarrow E_j(n) - e_{jk}$;
   - **For** each $(j,k) \in L(n+1)$ do:
     - **If** $E_j(n+1) < e_{jk}$
       - **Then** remove (j,k) from $L(n+1)$
5. Remove all *isolated* nodes from $V(n+1)$ and all its incoming and outgoing links from $L(n+1)$.
6. **For** each $j \in p$ do:
   - $E_j(n+1) \leftarrow E_j(n)$.

The pseudo-code of a min-max routing algorithm that utilizes cost function C_{jk} looks the same, except that it does not use the regular shortest path algorithms like Dijkstra or Bellman-Ford shortest path algorithms. As stated above, one can use a slightly modified Bellman-Ford algorithm.

3 Performance Evaluation

In the previous section, we have introduced a number of routing algorithms that use energy-aware routing metrics. The next question is to find a good scheme to evaluate the performance of the various algorithms and to compare them. Only in very few and simplistic cases it is possible to perform an analytical mathematical analysis of the algorithms. Therefore, we have performed extensive simulations that allow us to give a proper answer to this problem.

The other big challenge is to select a good criterion for comparison of the algorithm performance. No universal criterion that is suitable for all possible applications exists. Different applications have quite different requirements, so the criterion suited for some kind of missions might be useless for others. Table 3.1 contains a list of criteria, found in literature and used for evaluation and comparison of the performance of different routing algorithms. We now review that list.

The most popular criterion in the literature is *lifetime*. Two quite different definitions of lifetime can be found in the literature. According to one criterion, the *network lifetime* is the period until the first network node depletes its batteries' resources and hence becomes dead. In other words, network lifetime is defined as *the time of the first node failure* due to battery. The definition above can be found in several sources, such as [1], [2], [10]. In addition, there are a number of sources, such as [4], [6], [7], [8], [9], where a very similar criterion is used in order to evaluate the system vitality, although it is not called "lifetime". Recall that a node is said to be *isolated* if there are no active nodes in its transmission range. Note that no node can distinguish between a *dead* neighbor with a completely discharged battery and an *isolated* one that still has a certain amount of energy and thus is able to receive several more packets. Therefore, it seems rational to combine these two inactive states of a node into one by declaring a node as *dead* as soon as it becomes *isolated*.

A second definition of lifetime appears in [3], [5] as the period until there is some packet that cannot be delivered to its destination. In order to avoid ambiguity, we shall use here the term – *time of first packet loss* – instead of lifetime. One can intuitively conclude that there is certain dependence between these two definitions, although they are definitely not equivalent. When the event described in the first definition (*first node failure*) occurs, it does not necessarily mean that some kind of "system failure" happens. For example, one can think of a situation when some active node j exhausts its battery, turns off its equipment and disappears from the network. However, there are other nodes in its near environment, which can fulfill j 's duties, so the node's disappearance is almost unnoticed by other nodes. Note that a sensor network might consist of a large number of nodes and maybe only part of them is actually needed for successful network transmissions. The remaining nodes may be used in order to improve the network reliability. Most of the time they may be in the energy saving (sleep) state and wake up from time to time in order to check if their active participation is desired. This discussion shows that there might be situations when the first node failure may not be very critical.

On the other hand, the *first packet loss* is a much more critical event. It shows that from now on, packets may be rejected because of lack of energy.

The next two criteria – *expiration time* and *expiration sequence* – extend the concept of lifetime of an individual node to all nodes in the network. The *expiration sequence* is the sequence of nodes sorted in increasing order of their *expiration time* [4], [6], [7]. The comparison based on this criterion appears to be difficult (see Figure 3.1), since it is not clear how to compare numerical sequences.

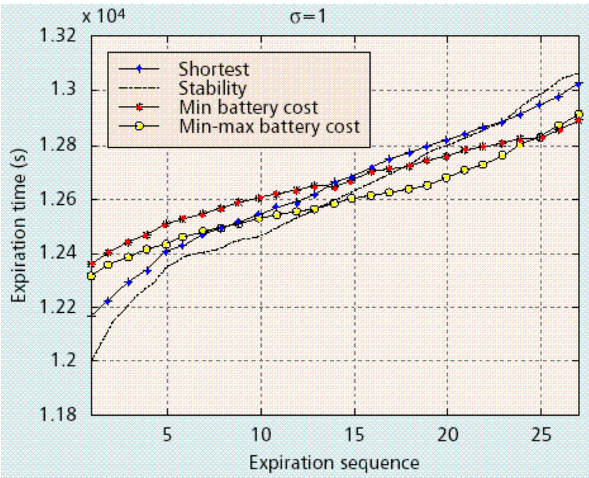


Figure 3.1. The figure was taken from [4], showing the expiration sequence of four algorithms. The expiration sequence is the sequence of nodes sorted in increasing order of their expiration times [6]. Consider two pairs of algorithms: "Min battery cost" (MBCR) vs. "Min-max battery cost" (MMBCR) and "Min battery cost" (MBCR) vs. "Shortest". Since almost all points of the red curve (MBCR) are located above the corresponding points of the yellow curve (MMBCR), the MBCR algorithm can unambiguously be declared as a winner in the first pair. It is not clear which algorithm of the second pair is preferable.

Another criterion widely proposed in the literature is *network throughput*. Both the *average* [8], [9] and the *total* [3], [6] *throughput* are used for performance evaluation and comparison of routing algorithms. The *average throughput* is defined as the rate of data received at the nodes, while the *total network throughput* (*capacity*) is the number of packets successfully delivered to their destinations [3], [6], [8], [9], until no more packets can be delivered. Undoubtedly, this measure is very important for analysis and comparison of performance in wired networks.

Observe that when measuring throughput, we measure only the *number of delivered packets*, and not the *efficiency of service* provided by the algorithm. For example, we do not ask questions like "How many packets were rejected until the last successful packet was delivered to its destination?" Intuition suggests and the results received from simulations show, that the number of packets rejected by all algorithms discussed in the previous chapter is vastly larger than the total throughput achieved by them. This fact can be explained as follows. We design some connected network, define a packet sequence, and run some algorithm. The strongly connected network we start with eventually splits into several disconnected sub-networks, because of battery depletion. After a certain time, each of the sub-networks splits as well. At this stage, some packets can still be delivered to their destinations, but most packets are rejected. It takes a long time until absolutely no more packets can be delivered. Therefore, the *throughput* criterion hardly gives a good indication of the actual network performance.

From the above discussion it appears that a more general measure of the algorithm performance than the ones currently used in the literature - *lifetime* and *throughput* - is required. Measuring only *throughput* takes us too late in the progress of the network behavior, the time when absolutely no packet can be sent. The other extreme is when *lifetime* is measured, namely the time when the first packet is rejected. Of course, the first packet loss is an important event. On the other hand, given the nature and possible size of sensor networks, it seems that this event is not so critical and loss of a certain percentage of packets is tolerable.

In order to accommodate these thoughts, we introduce the terms *packet delivery ratio* at time t, *packet loss ratio* at time t and *threshold-related throughput*. The number of packets offered to the network until time t is the sum of delivered and rejected packets up to that time. The packet delivery ratio at time t, denoted by $\sigma(t)$, is defined as the number of delivered packets up to time t divided by the number of packets offered to the network up to that time. The packet loss ratio at time t, denoted by $\rho(t)$, is defined as the ratio of the number of rejected packets and the number of offered packets up to time t. Obviously, $\rho(t) = 1 - \sigma(t)$. The *threshold-related throughput* corresponding to some *threshold* Θ, $0 \le \Theta \le 1$, is defined as the total number of delivered packets until the first time when the packet loss ratio exceeds threshold Θ.

With these definitions, one can easily see that Lifetime is the threshold-related throughput corresponding to $\Theta = 0$, while Total Network Throughput is the threshold-related throughput corresponding to Θ close to 1, since loss of any number of packets is permitted. In order to thoroughly evaluate algorithm performance, one should measure the threshold-related throughput for a wide range of threshold values. *Lifetime* and *Total Throughput* are not sufficient. To provide a specific example, it is asserted in many references (e.g. [1], [2], [3], [4]) that the *lifetime* (time of first node failure or first packet loss) of the MTER (Minimum Transmitting Energy Routing) algorithm is significantly less than that of other schemes, and therefore the other scheme is preferable. This conclusion does not take into account that for any non-zero value of Θ, the MTER algorithm obtains better *threshold-related throughput* (see Sec. 4 below), which makes it more attractive than the other scheme.

Appendix 3.A –Table of Comparison Criteria

Name	Description	References
Lifetime (time of first node failure)	[1,2]: The length of time until the first battery drains out . [10]: The time elapsed from the time instant when all nodes have a fully charged battery to the time instant when the first node in the network runs out of battery. [8,9]: The time when the first node dies.	[1], [2], [7], [8], [9], [10], [13]
Time to network partition (lifetime)	[8,9]: A routing algorithm, which minimizes the standard deviation of nodes' lifetime, is predictable and thus desirable.	
Average and standard deviation of nodes' lifetime	[8,9]	
Time for last node to die	[8,9]	
Lifetime (time of first packet loss)	[3]: The number of packets successfully routed until the first rejection. [5]: The earliest time when a packet cannot be sent due to saturated nodes.	[3], [5]
Expiration time and expiration sequence	[4]: The time when a node exhausts its battery capacity. [6]: The sequence of nodes sorted in ascending order of the expiration times. The expiration sequence provides a useful indicator of how each algorithm affects the lifetime of the individual nodes, and the entire network. [7]: No strict definition of that term was found in [7]. The expiration sequence, namely the number of time slots until the first and the last node dies, is used in order to evaluate lifetime.	[4], [6], [7]
Maximum network capacity Total packet throughput	[3]: The total number of packets successfully routed until no more packets can be routed. [6]: The total number of packets successfully received at the destination nodes.	[3], [6]

Network throughput	[8,9]: The rate of data packets received at the gateway.	[8], [9]
Throughput until *m* packet delivery failure	[5]: The results can be relaxed to accommodate up to *m* packet delivery failures, with *m* a constant parameter.	[5]
Average delay per packet (latency)	[8,9]: The average time a packet takes from a node in the sensing state to the gateway. [10]: The average delay from the time instant when a packet is generated at the source node to the time instant when it is delivered to the destination.	[8],[9], [10]
Average energy consumed per packet	[8,9]: The average energy consumed in transmitting and receiving a data packet. A routing algorithm that minimizes the energy consumed per packet will, in general, yield better energy savings and increased network lifetime. [6]: The total energy expenditure divided by the total packet throughput.	[6], [8], [9]
Total energy dissipated in the system	[7]	[7]
Frequency of receiving data packets at the end nodes	[14]: In evaluating performance, researchers used a program that records how often data is being received through the gateway nodes.	[14]
Threshold-Related Throughput	The total number of delivered packets until the first time when the loss ratio exceeds some given threshold	This work

Table 3.1. Criteria for evaluation and comparison of algorithms' performance.

4 Simulations

A number of algorithms discussed above have been selected in order to evaluate their performance with respect to a number of comparison criteria. Performance of algorithms was widely studied under different conditions.

For each simulation scenario, we generated ten different networks and for each network we have produced ten different packet sequences. The originating and the target nodes of each packet are uniformly distributed among all appropriate sources and destinations. All algorithms perform a single shortest path computation for each packet. A discrete (slotted) time system was used for the sake of simplicity. We assume a packet is produced at the beginning of each time slot (round). That packet is delivered during the current time slot. If no feasible path from source to destination can be found, the packet is rejected. Regular and modified versions of the Bellman-Ford algorithm (see Chapter 2) have been used in the simulations in order to compute minimum and min-max paths respectively.

For the CMAX algorithm we experimented with different values of λ and found that the algorithm performance is relatively insensitive to the value of λ, as long as it is large enough, so λ = 100,000 was used.

The following parameters were measured for each of the compared algorithms:

1. *Lifetime*: the time slot in which the first node becomes isolated (dead);
2. *Total energy dissipated in the system*: the total energy consumed by all nodes during the execution of the algorithm;
3. *Average energy consumed per packet*, namely the total energy dissipated in the system divided by the total network throughput (capacity);
4. *The total network throughput* (*capacity*);
5. *The threshold-related throughput* as a function of the *threshold* Θ, namely the number of packets successfully delivered to their destinations until the first time when the loss ratio exceeds Θ. The loss ratio $\rho(n)$ is computed as the number of rejections divided by the current time slot n. Note that n is also the total number of packets generated until that time, since exactly one packet is generated in each slot. Ten values from 0 to 0.9 with a step of 0.1 were selected for Θ. Note that the throughput measured for $\Theta = 0$ (rejections are not allowed at all) and Θ close to 1 (rejection of any number of packets is allowed) represent the

throughput until first packet rejection (lifetime) and the total network throughput (capacity) respectively.

6. *The total number of rejected (lost) packets*, namely the number of packets rejected until the last successful one was delivered to its destination.

4.1 Performance Evaluation

In this set of simulations, we have studied the performance of several routing algorithms for homogeneous sensor networks, consisting of nodes with identical initial battery power. The performance of the following algorithms was studied:

1. MTER (Minimum Transmitting Energy Routing): $C_{jk}(t) = e_{jk}$;
2. MBCR (Minimum Battery Cost Routing): $C_{jk}(t) = 1/E_{jk}(t)$;
3. MREPcapsum (Minimum Residual Energy Path): $C_{jk}(t) = e_{jk}/E_{jk}(t)$;
4. MREPsum: $C_{jk} = 1/(E_j(t) - e_{jk})$;
5. CMAX (Capacity MAXimization): $C_{jk}(t) = e_{jk} \cdot (\lambda^{a_j(t)} - 1)$, where $a_j(t) = 1 - E_j(t)/E_j$;
6. Modified CMAX (MCMAX): $C_{jk}(t) = e_{jk} \cdot \lambda^{a_j(t)}$,where $a_j(t) = 1 - E_j(t)/E_j$;
7. MMBCR (Min-Max Battery Cost Routing): $C_{jk}(t) = 1/E_{jk}(t)$;
8. MREPmax: $C_{jk} = 1/(E_j(t) - e_{jk})$.

The first six algorithms use minimum cost and the last two algorithms use min-max cost methods in order to find an appropriate route for a given packet.

4.1.1 Scenario I

The network consists of fifty nodes, randomly distributed on a 40×40 square-units area. In this scenario, we want to evaluate the algorithms' performance under most unrestrictive conditions: packets are generated between all possible source-destination pairs; neither the sequence of future packets nor the origination rates of the packets are known in advance.

The upper bound on *total network throughput* has been derived by us. It can be found in the Theorem 2 (see Appendix 4.A). Unfortunately the bound cannot be reached by any *online* routing algorithm for an arbitrary generated packet sequence.

The energy consumed by node j in transmitting a unit length packet to a neighboring node k is computed according to the following formula: $e_{jk} = max\{0.001, 0.001 \times d_{jk}^3\}$, where d_{jk} is the physical distance between the nodes. A similar energy consumption model was used in [1], [2], [3], [5]. We assume that the maximal transmitting distance of a node is determined by its energy resources. Each node can directly transmit to each other node within the area if its residual energy allows it. Three values of the initial energy, 8, 15 and 30 units, were used. A sample network is shown on Figure 4.1. One can see how the initial neighborhood of a node changes depending on the value of the initial energy assigned to it.

Nodes are declared *dead* as soon as there are no active neighbors in their transmission range. It is assumed that packets are not allowed to be sent to any dead node; therefore, if there is such a packet at the current time slot, it is discarded.

Each of the above-mentioned algorithms was run 300 times: ten different packet sequences were simulated in each of ten randomly generated networks for each of the three possible values of the nodes' initial energy (8, 15 and 30 units). The averaged results are presented later in this section.

The following conclusions were drawn from the simulation results:

1. The criterion *"total amount of energy dissipated in the network"* hardly gives any indication of the actual network performance.
2. Using *"lifetime"*, *"time of first packet loss"* or *"total throughput"* as comparison criteria yields sometimes to incorrect conclusions.
3. The *"threshold-related throughput"* criterion proposed in this work provides a much better indication of the algorithms behavior.
4. Using the *"threshold-related throughput"* criterion, we have evaluated the performance of and have compared a variety of routing algorithms. The best performance was obtained by the MREPcapsum routing algorithm for a wide range of network topologies, packet generation sequences and initial node energy values.

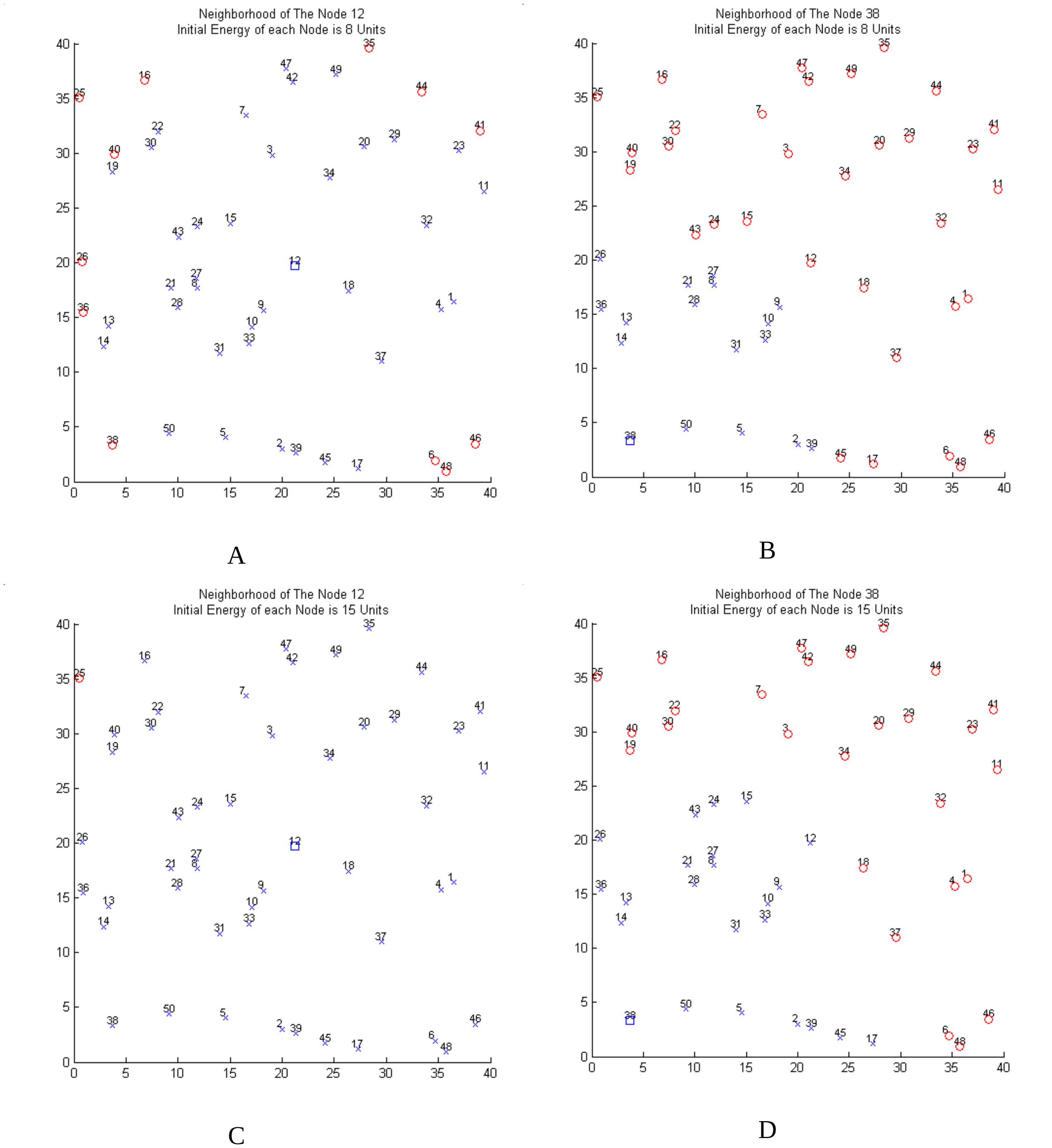
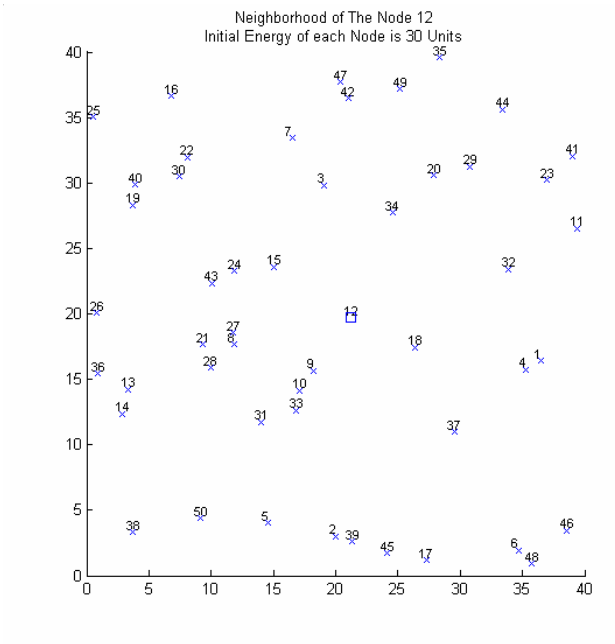
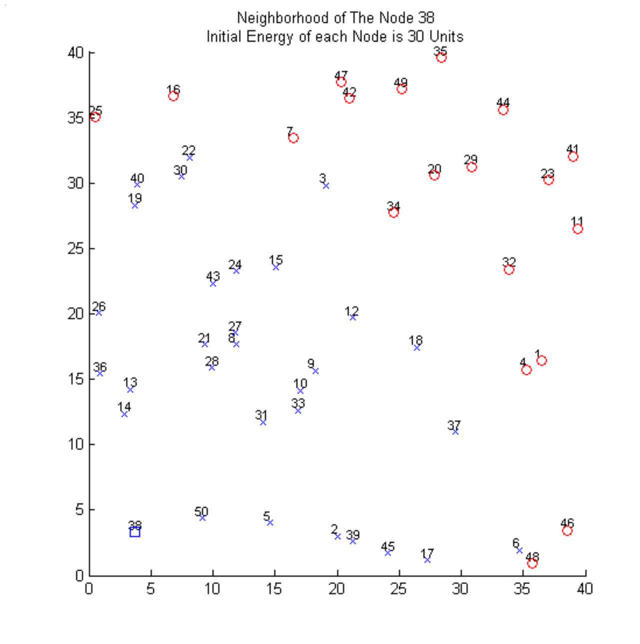


Figure 4.1.A-D. Sample network consisting of 50 nodes deployed on a 40×40 area. The initial energy of each node is 8 units (Figures 4.1.A, 4.1.B) or 15 units (Figures 4.1.C, 4.1.D). Neighborhoods of the sample nodes 12 and 38 (see blue squares) are shown on figures 4.1.A, 4.1.C and 4.1.B, 4.1.D respectively. The nodes that are within the communication range of the sample nodes are drawn as blue crosses. The ones outside this range are drawn as red circles.



E



F

Figure 4.1.E-F. Sample network consisting of 50 nodes deployed on a 40×40 area. The initial energy of each node is 30 units. Initial neighborhoods of the sample nodes 12 and 38 (see blue squares) are shown on figures 4.1.E and 4.1.F respectively. The nodes that are within the communication range of the sample nodes are drawn as blue crosses. The ones outside this range are drawn as red circles.

Results

The averaged results received from simulations can be found in Figures 4.2 – 4.10 and in Table 4.1 below. The results obtained in each given simulation are similar to the average results.

The *threshold-related throughput* measured for ten different values of *loss ratio threshold* is depicted in Figures 4.2 – 4.4. One can see that as the initial energy of the nodes is multiplied by some factor, the throughput of almost all algorithms (except MBCR and its min-max version MMBCR) is multiplied by the same factor. For small values of initial energy (8 and 15 units), the MREPcapsum algorithm yields the best results in terms of *threshold-related throughput*, measured for any *loss ratio threshold*. When the initial energy of nodes is 30 units, both the MREPcapsum and MCMAX routing algorithms yield very similar results and outperform all other algorithms. Note that the *first packet loss* always occurs earlier in the MTER algorithm than in the MREPmax algorithm (see Fig. 4.5, 4.6). However, for any given positive value of the loss ratio threshold, MTER achieves larger throughput (see Fig. 4.2 – 4.4). This shows that the *first packet loss* criterion does not give sufficient indication of the algorithm performance.

A similar situation can be observed in Figure 4.3. The CMAX routing algorithm achieves significantly better performance than the MREPsum algorithm in terms of *threshold-related throughput*, measured for any positive value of *loss ratio threshold*. However its *lifetime* is shorter than that of MREPsum (see Table 4.1). One can conclude that the *lifetime* does not provide reliable indication of actual algorithm performance.

Note that in the given scenario, death of first node *(minimal node lifetime*) and *first packet loss* are most of the time very close events (see Fig. 4.5 and 4.6, Table 4.1), so any one of them is sufficient for comparison purposes. However, as indicated above, these criteria do not provide sufficient information of the algorithm behavior.

Observe that in two simulations (initial energy of 8 and 15 units) the MTER algorithm achieves the maximal *total network throughput* among all compared algorithms (Figure 4.7). On the other hand, both the MREPcapsum and the MCMAX routing algorithms achieve better performance in the terms of *threshold-related throughput*, measured for a wide interval of *loss ratio threshold* values.

In addition compare Fig. 4.7 with Fig. 4.8. One can see that the *total number of packets rejected* by all compared algorithms is vastly larger than their *total throughput*. Thus measuring only *throughput* takes us too late in the progress of network behavior, to the time when absolutely no packet can be sent. This is an additional indication that the *threshold-related throughput* criterion appears to be a much more indicative, because it provides a much accurate measure of the algorithm behavior during its execution.

The conclusion based on measurement of *total energy dissipated in the system* during algorithms execution is that all evaluated algorithms almost completely spend all available energy (see Fig. 4.9). One can see that no reasonable conclusion can be reached by attempting to compare the algorithm performance using this criterion.

In [8] and [9] it was asserted that "a routing algorithm that minimizes the energy consumed per packet will, in general, yield better energy savings and increased network lifetime". Our simulations show that this is not necessarily correct. In two simulations (initial energy of 8 and 15 units), the *average energy of delivering a packet* using the MTER algorithm has minimal value among all studied algorithms (see Figure 4.10). However, MREPcapsum, MCMAX and even MREPmax yield much better results with respect to both *minimal node lifetime* and *first packet loss* (see Figures 4.5 and 4.6).

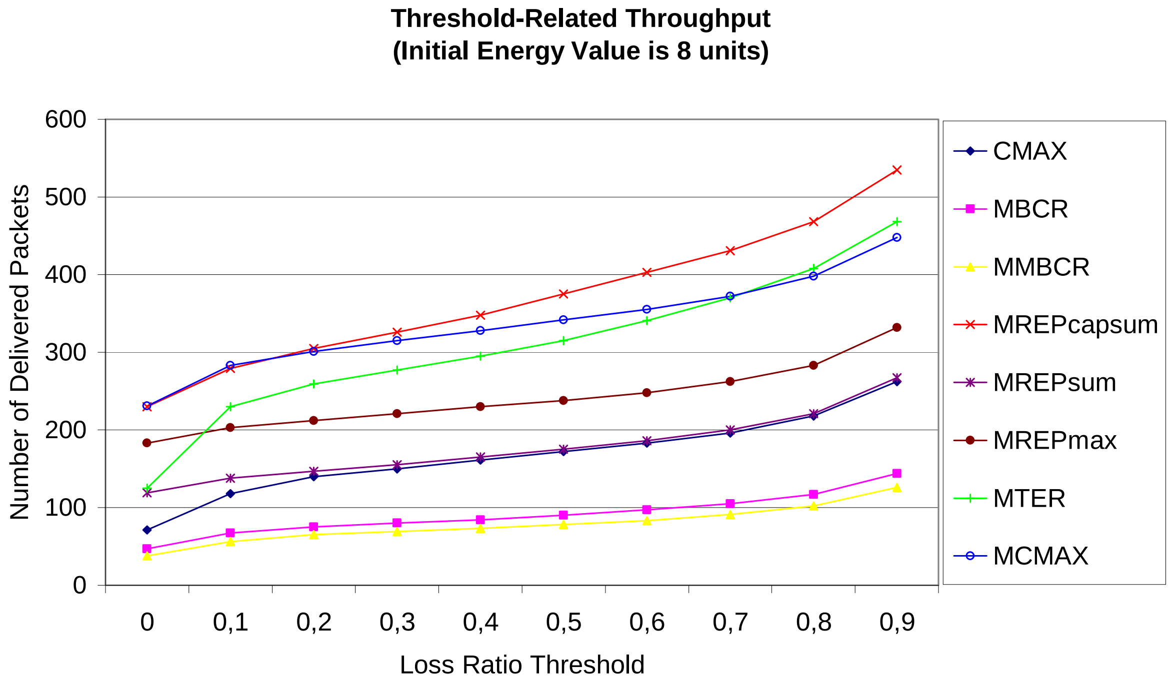


Figure 4.2. Threshold-related Throughput vs. Loss Ratio Threshold.

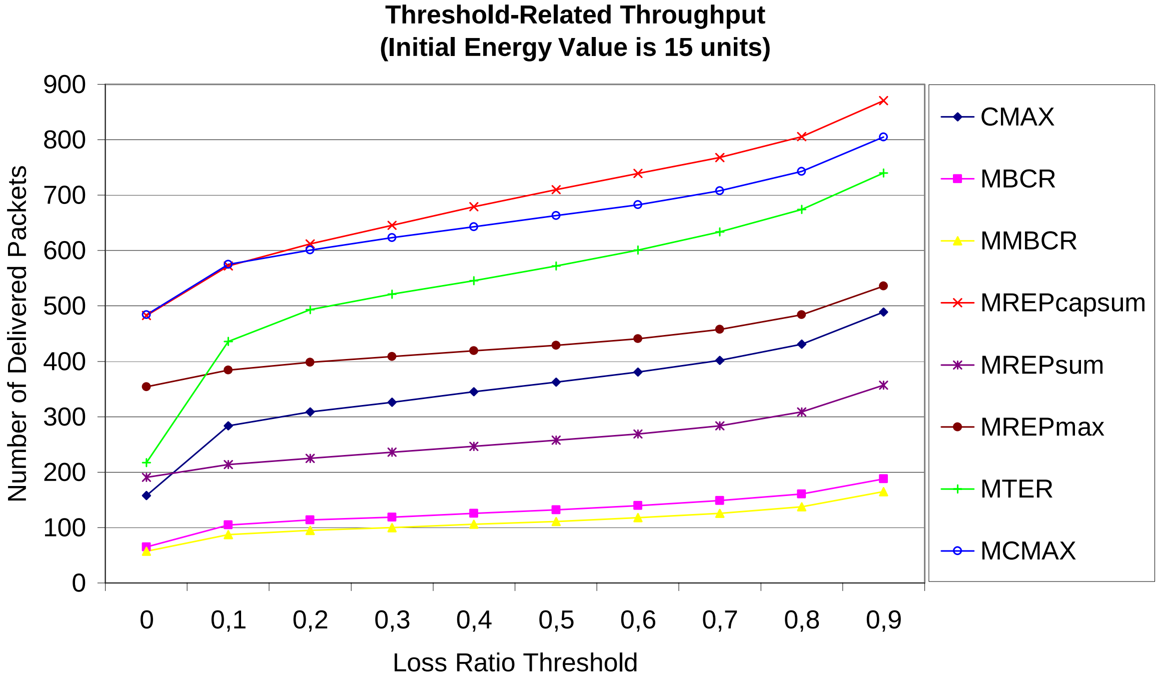


Figure 4.3. Threshold-related Throughput vs. Loss Ratio Threshold.

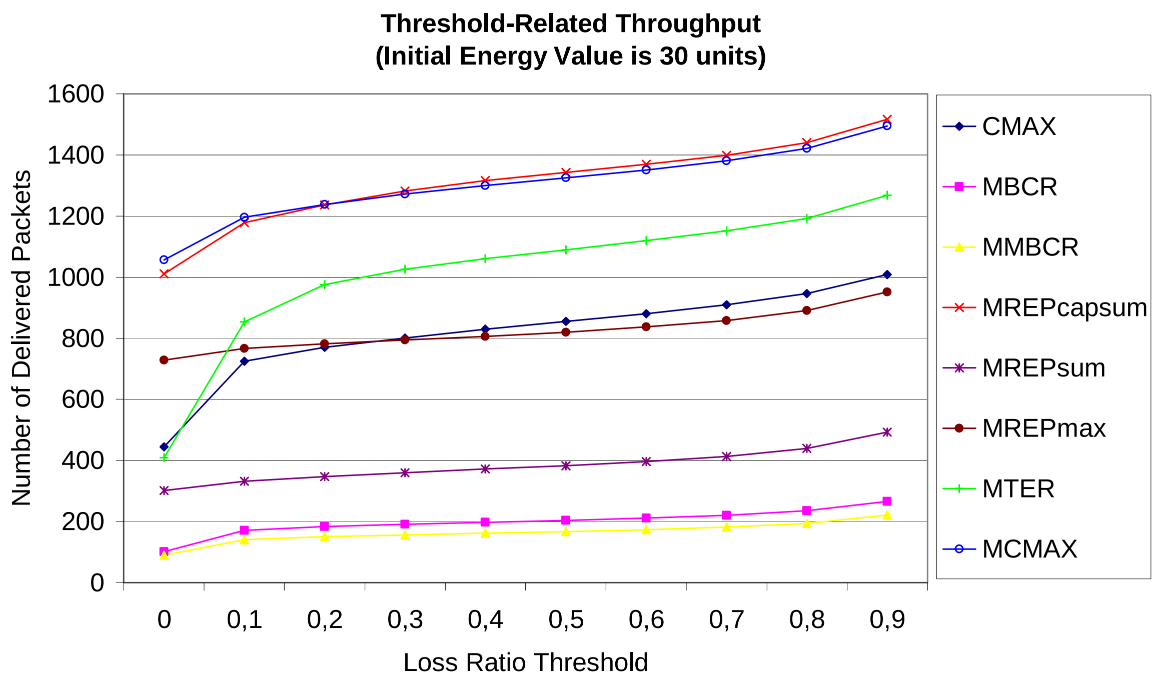


Figure 4.4. Threshold-related Throughput vs. Loss Ratio Threshold.

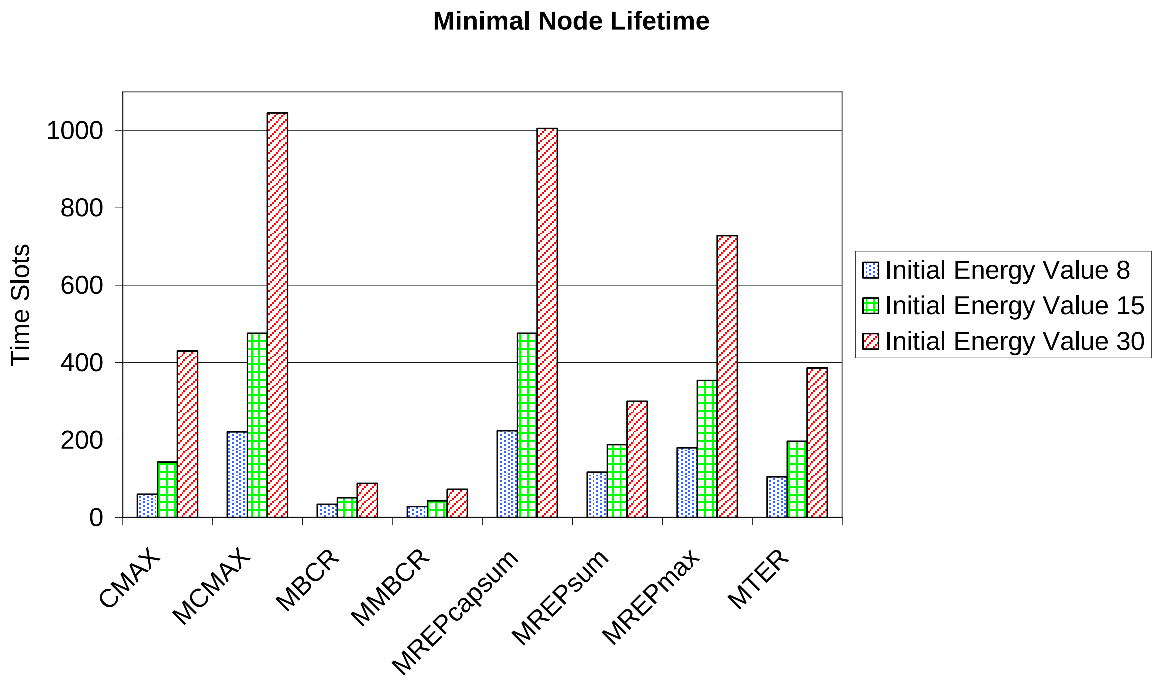


Figure 4.5. Minimal Node Lifetime.

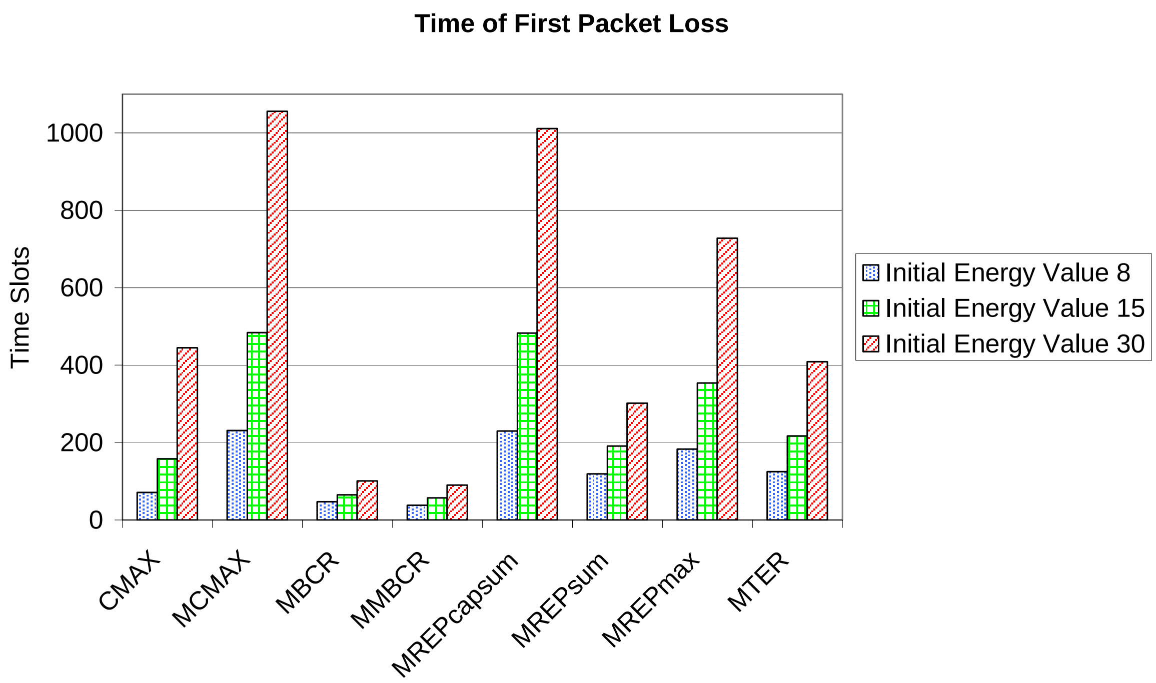


Figure 4.6. Time of First Packet Loss.

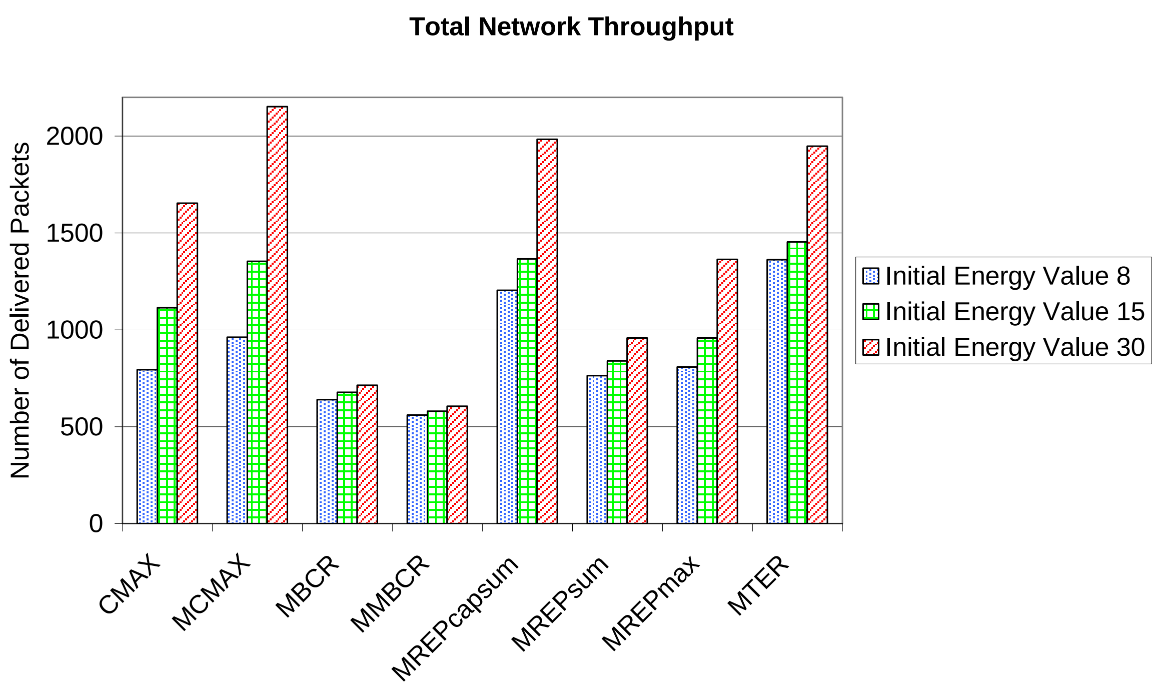


Figure 4.7. Total Network Throughput (Capacity).

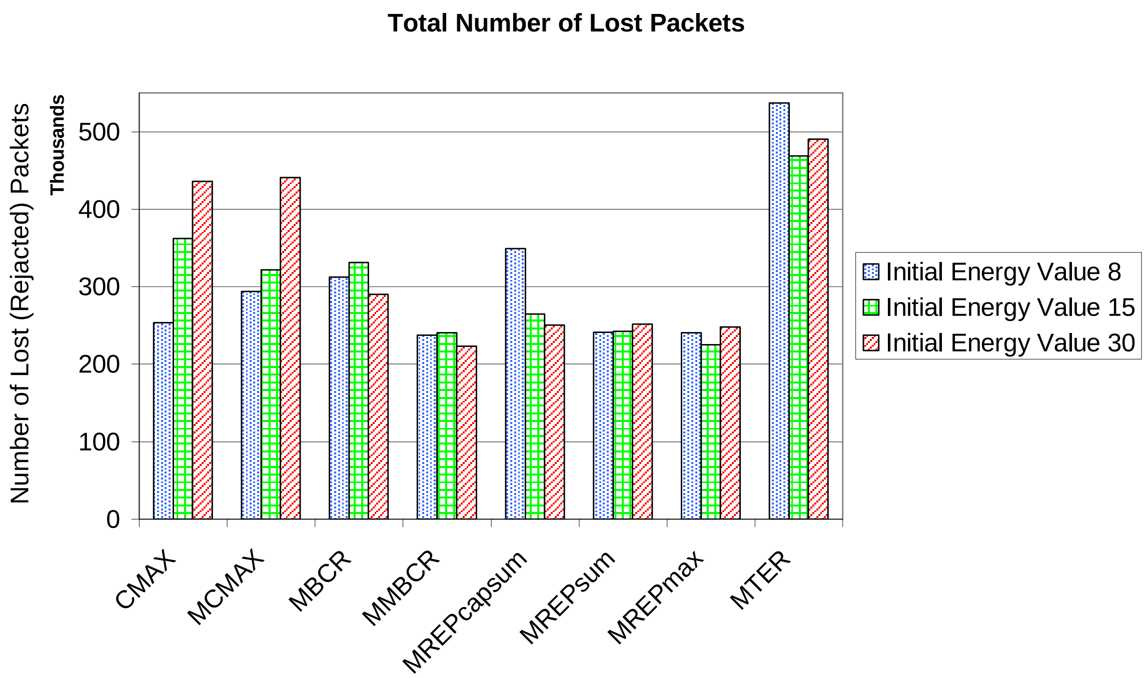


Figure 4.8. Total Number of Lost Packets.

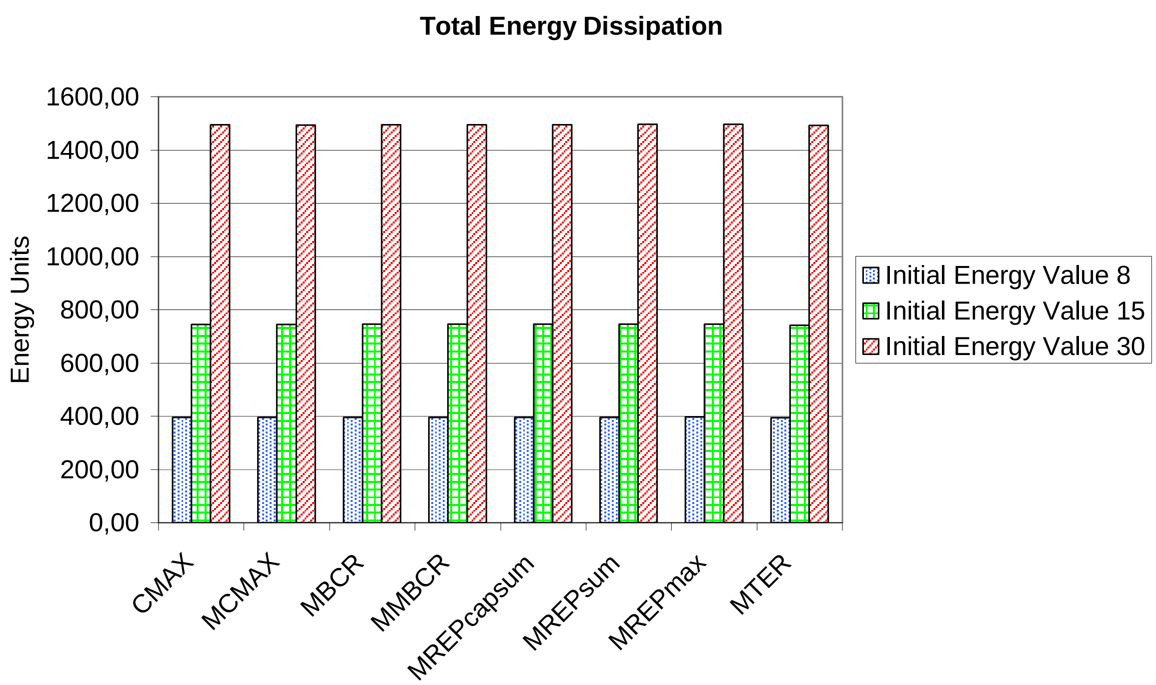


Figure 4.9. Total Energy Dissipated in System during Algorithms Execution.

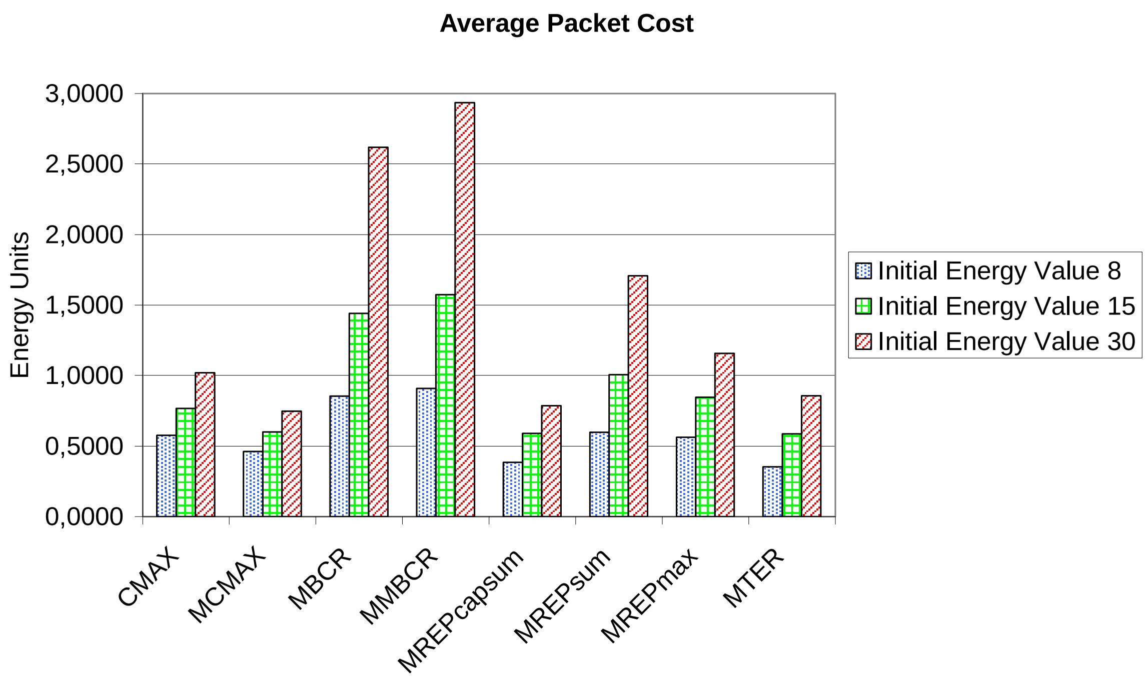


Figure 4.10. Average Cost of Delivering Packet.

	Lifetime	Time of first packet loss	Total throughput	Total num. of lost packets	Total en. diss.	Av. packet cost
Initial energy value of the node – 8 units						
MTER	105	125	1363	536872	393.84	0.3526
MBCR	34	47	641	312531	395.74	0.8533
MREPcapsum	224	230	1204	349355	395.94	0.3850
MREPsum	117	119	765	241156	396.31	0.5969
CMAX	60	71	794	253899	395.57	0.5761
MCMAX	221	231	962	293985	395.63	0.4607
MMBCR	28	38	561	237453	395.70	0.9082
MREPmax	180	183	809	240733	396.39	0.5612
Initial energy value of the node – 15 units						
MTER	197	217	1455	468513	742.49	0.5864
MBCR	51	65	678	331085	745.87	1.4392
MREPcapsum	476	483	1366	264899	745.79	0.5885
MREPsum	188	191	840	242599	746.39	1.0071
CMAX	143	158	1114	362259	745.23	0.7663
MCMAX	479	484	1355	321554	745.15	0.6005
MMBCR	43	57	581	240620	745.94	1.5733
MREPmax	354	354	959	225207	746.40	0.8457
Initial energy value of the node – 30 units						
MTER	386	409	1948	490213	1491.70	0.8557
MBCR	88	101	714	289911	1496.00	2.6171
MREPcapsum	1005	1011	1984	250334	1495.80	0.7857
MREPsum	300	302	958	251868	1496.20	1.7061
CMAX	430	445	1655	435738	1494.80	1.0204
MCMAX	1045	1056	2152	440977	1494.50	0.7469
MMBCR	73	90	607	223238	1495.90	2.9342
MREPmax	728	728	1364	248073	1496.30	1.1571

Table 4.1. The average results received in the first simulation scenario. Each value is obtained by averaging the results of one hundred simulations.

4.1.2 Scenario II

In this part, we examine the performance of routing algorithms in a different configuration that appears often in ad-hoc, disaster recovery and sensor networks (Fig. 4.11). Fifty nodes are randomly distributed on a 40×40 square area. Four energy unconstrained sink nodes are posed outside of the area, at distance of 2 units from the area bound. All traffic is destined to these four sink nodes. Three different values of initial energy 8, 15 and 30 units were selected.

We used the same energy consumption and slotted time model as in the previous set. A packet is originated by a randomly selected source at the beginning of each time slot. During the current slot, this packet has to be delivered to any one of four possible destinations (sinks). Packet that cannot be delivered is rejected and its source is declared *dead*.

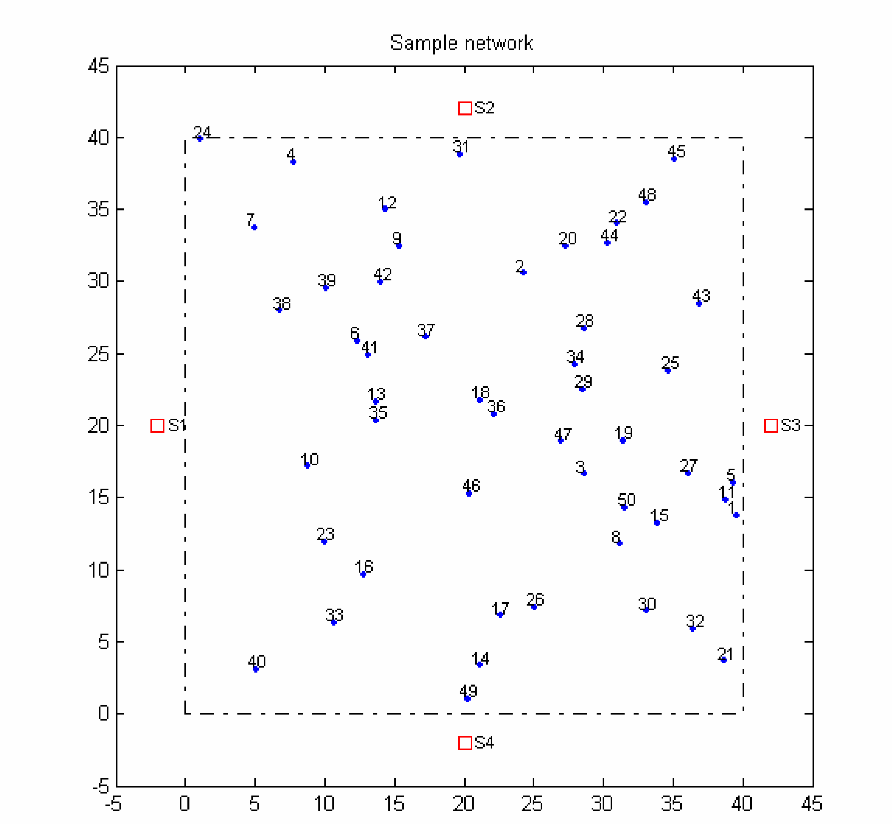


Figure 4.11. A sample network consisting of 50 regular energy-constrained nodes (blue points numbered from 1 to 50) deployed on 40×40 area is depicted in Figure 4.10. In addition, there are 4 energy-unconstrained sink nodes (red squares denoted by S_1, S_2, S_3, S_4) placed outside of the 40×40 area, not far from its borders. A node originating a packet is free to select any one of those four possible destinations in order to deliver the packet. Once a packet of some node cannot be delivered, the node is declared *dead*.

We have derived the upper bound on *total throughput* of such a network with a-priori known set of destination nodes. It can be found in Corollary 1 of Theorem 1 (see Appendix 4.A).

For this scenario, we developed an additional routing scheme, named MTTR, as follows. A node that generates a packet sends it directly, with no help from intermediate nodes, to the closest sink. Only if no direct transmission is possible to any of the sinks because energy is low, the packet is sent on a multi-hop route. The route is selected as the minimum transmitted energy route to any node from which direct transmission to a sink is possible. In Corollaries 2 and 3 of Theorem 1 (see Appendix 4.A), it is shown that for the used energy consumption model, the MTTR (Maximum Total Throughput Routing) routing algorithm reaches the derived bound on *total throughput* for any arbitrary generated packet sequence.

MTTR and each of the other algorithms previously mentioned was executed three hundred times: ten different packet sequences were simulated in each of ten randomly generated networks for three possible values (8, 15 and 30 units) of the nodes' initial energy. The main conclusions drawn for the previous simulation scenario are valid for this one as well:

1. The criterion *"total amount of energy dissipated in network"* hardly gives any indication of the actual network performance.
2. Using *"lifetime"*, *"time of first packet loss"* and *"total throughput"* as comparison criteria sometimes yields incorrect conclusions.
3. The *"threshold-related throughput"* criterion proposed in this work provides a much better indication of algorithm behavior.
4. Using the *"threshold-related throughput"* criterion, we have evaluated the performance of and have compared a variety of routing algorithms. The best performance was obtained by the MREPcapsum and the MCMAX routing algorithms for a wide range of network topologies, packet generation sequences and initial node energy values.

Results

The average results received from simulations can be found in Figures 4.12 – 4.16 and in Table 4.2 below. The results of each particular simulation are similar to the average results.

The *threshold-related throughput* measured for ten different values of the *loss ratio threshold* is depicted in Figures 4.12 – 4.14. One can see that for all values of initial energy both the MREPcapsum and the MCMAX routing algorithms yield the best performance among all compared algorithms. The MTER algorithm displays very similar results for any non-zero value of *loss ratio threshold*. However, as in the previous model, the *first packet loss* of the MTER algorithm occurs quite early.

Like in the previous simulation scenario, in most of the studied algorithms, network *lifetime* is doubled when the initial energy of the nodes is doubled. The exceptions are MBCR and its min-max version MMBCR, for which the *lifetime* is increased by a much smaller amount, and MREPmax, where doubling the initial energy of the nodes causes the *lifetime* to be increased by a factor of four (see Figures 4.15 and 4.16).

By comparing the *lifetime* of MTER algorithm vs. that of MREPsum (see Figures 4.15 and 4.16) and their *threshold-related throughput* measured for any positive value of *loss ratio threshold* one can see as before that *lifetime* does not provide sufficiently good information.

Note that the *total network capacity* achieved by MREPsum for all simulated values of initial energy is very close to the maximal value achieved by MTTR (see Figure 4.17). One can see that although the latter always achieves the upper bound on *total throughput,* its performance in the terms of *threshold-related throughput* is quite poor. In addition, observe that the *total number of packets rejected* by all compared algorithms is vastly larger than the *total throughput* achieved by them (Figure 4.18). Thus, *total throughput* measured at the time when absolutely no packet can be sent cannot be a good comparison criterion. The *threshold-related throughput* appears to be much more indicative.

Finally, like in the previous simulation set, not many conclusions can be drawn from a criterion like *total energy dissipation* (see Figure 4.19).

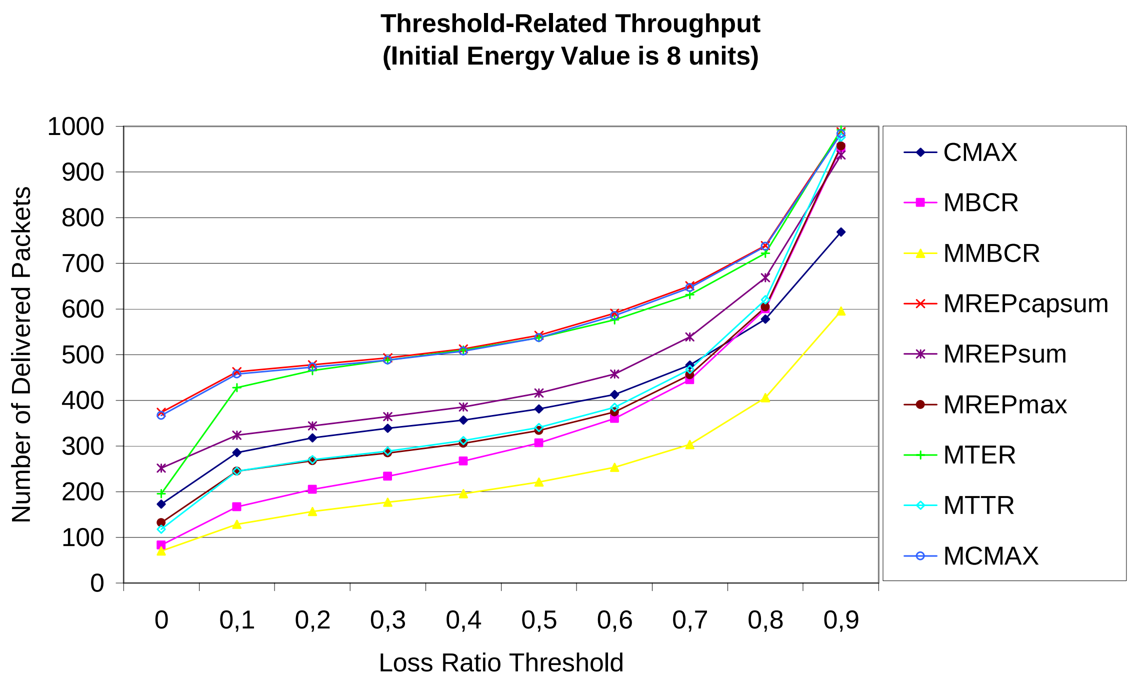


Figure 4.12. Threshold-related Throughput vs. Loss Ratio Threshold.

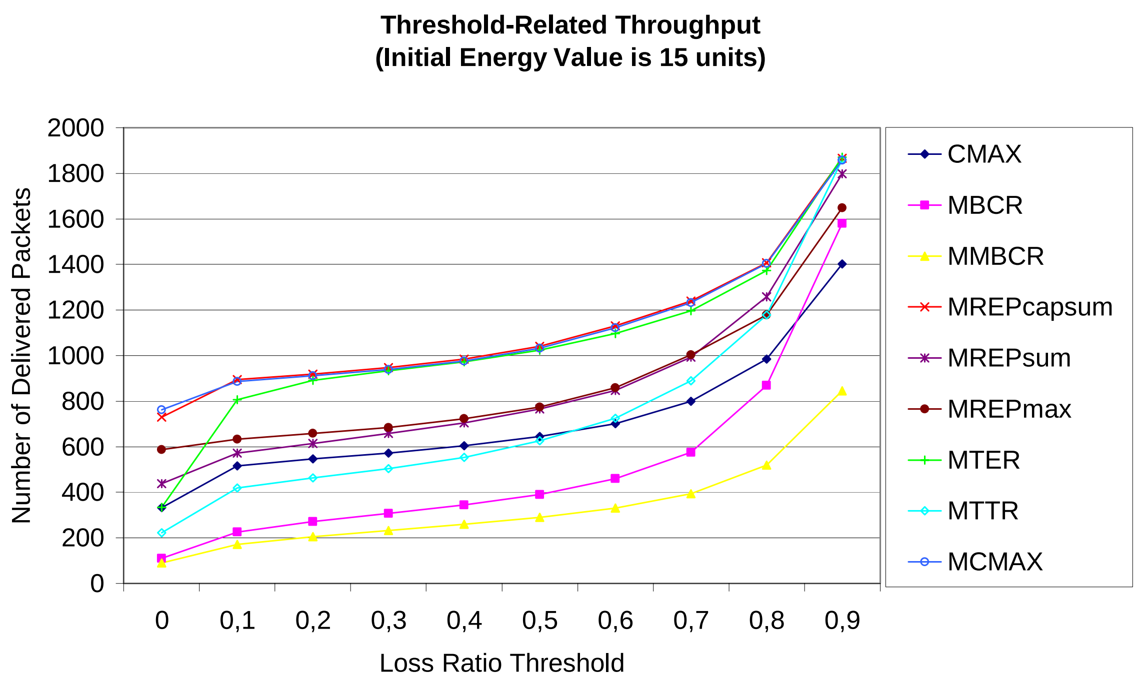


Figure 4.13. Threshold-related Throughput vs. Loss Ratio Threshold.

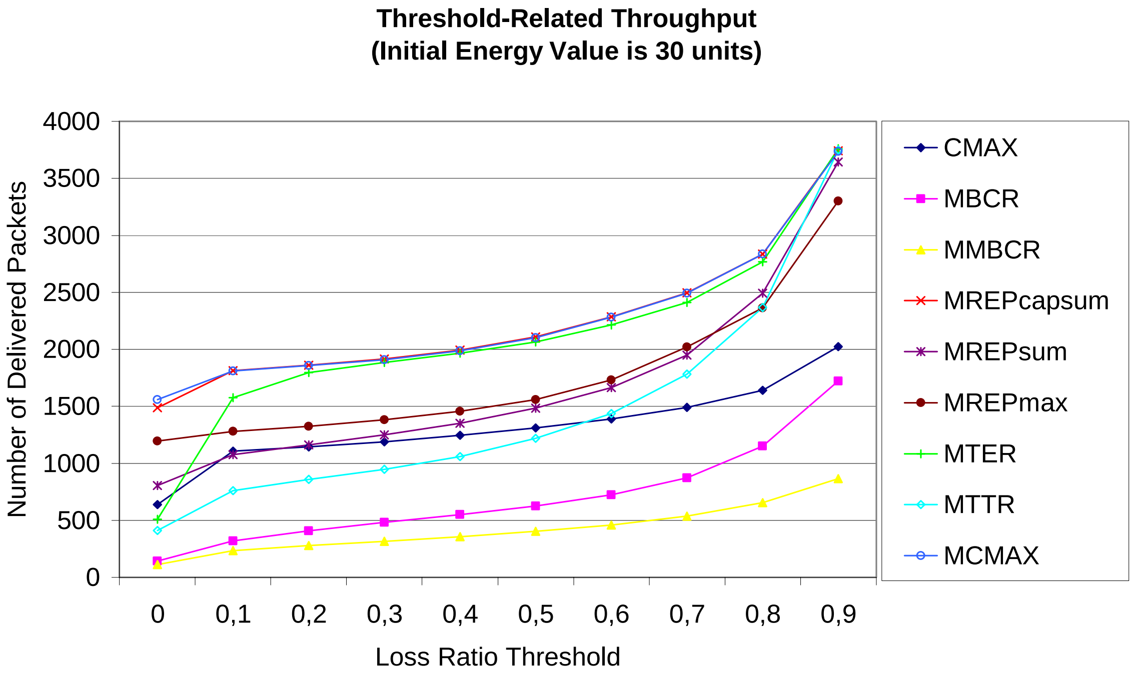


Figure 4.14. Threshold-related Throughput vs. Loss Ratio Threshold.

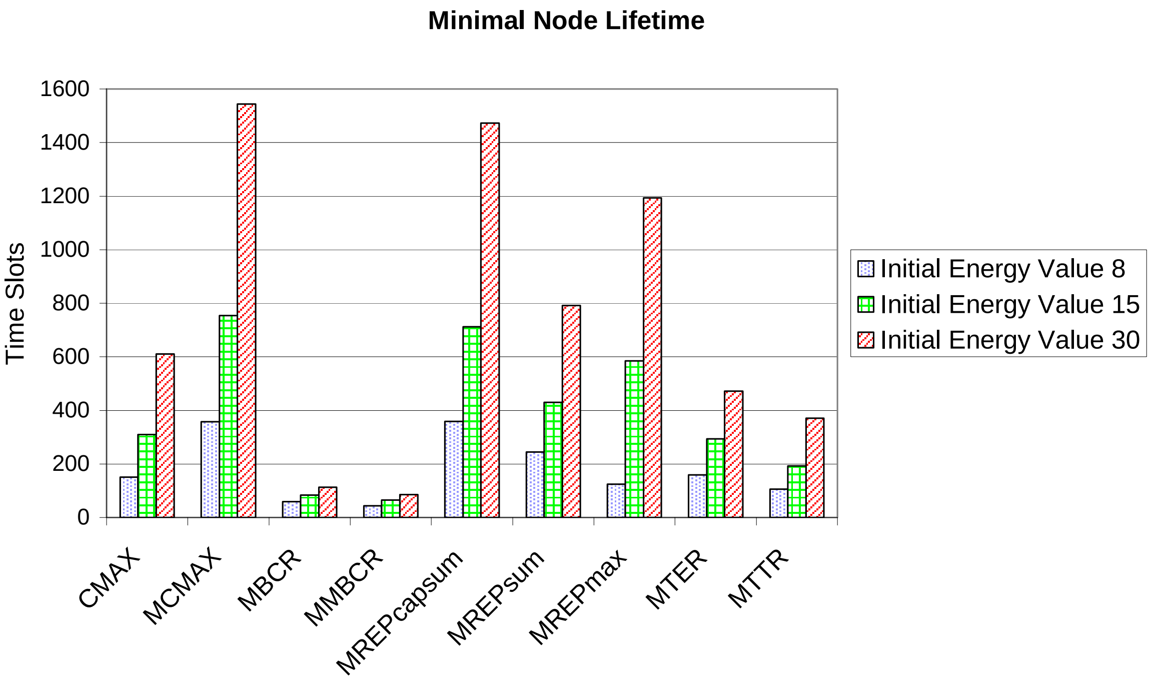


Figure 4.15. Minimal Node Lifetime.

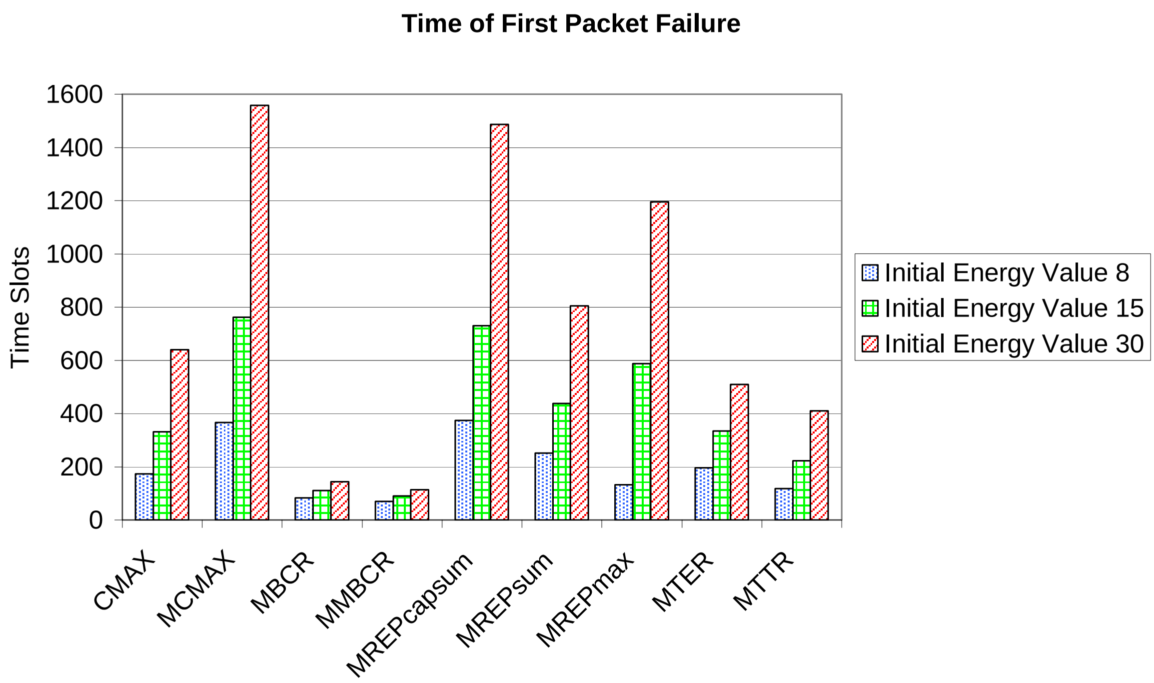


Figure 4.16. Time of First Packet Loss.

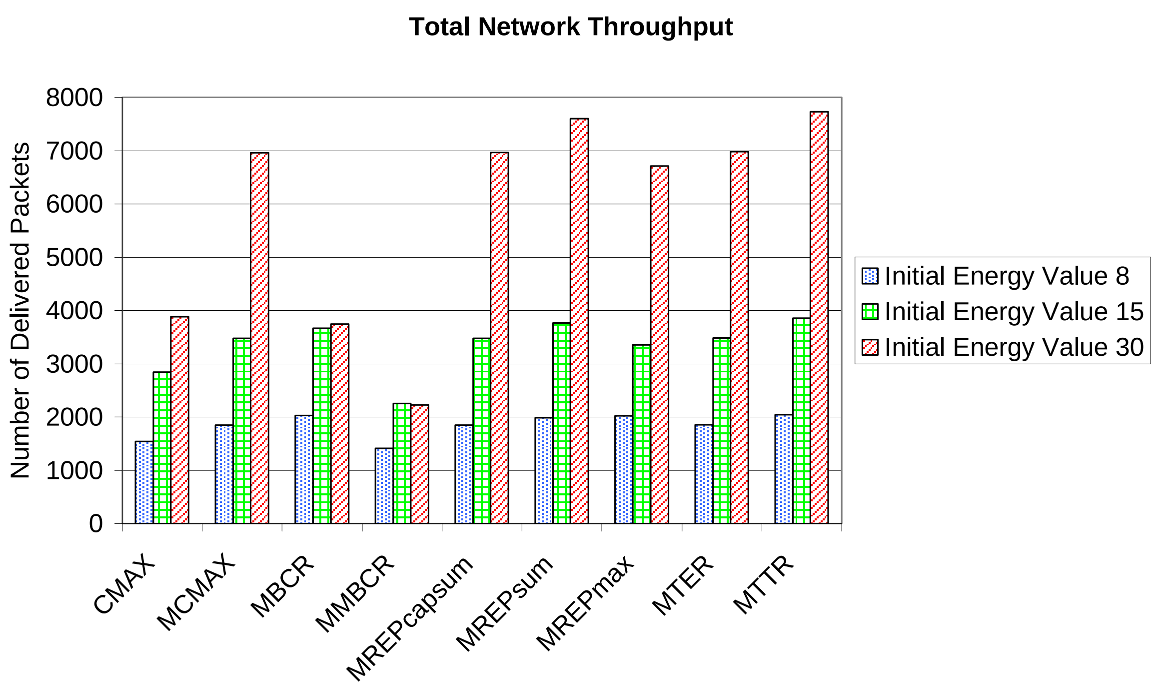


Figure 4.17. Total Network Throughput (Capacity).

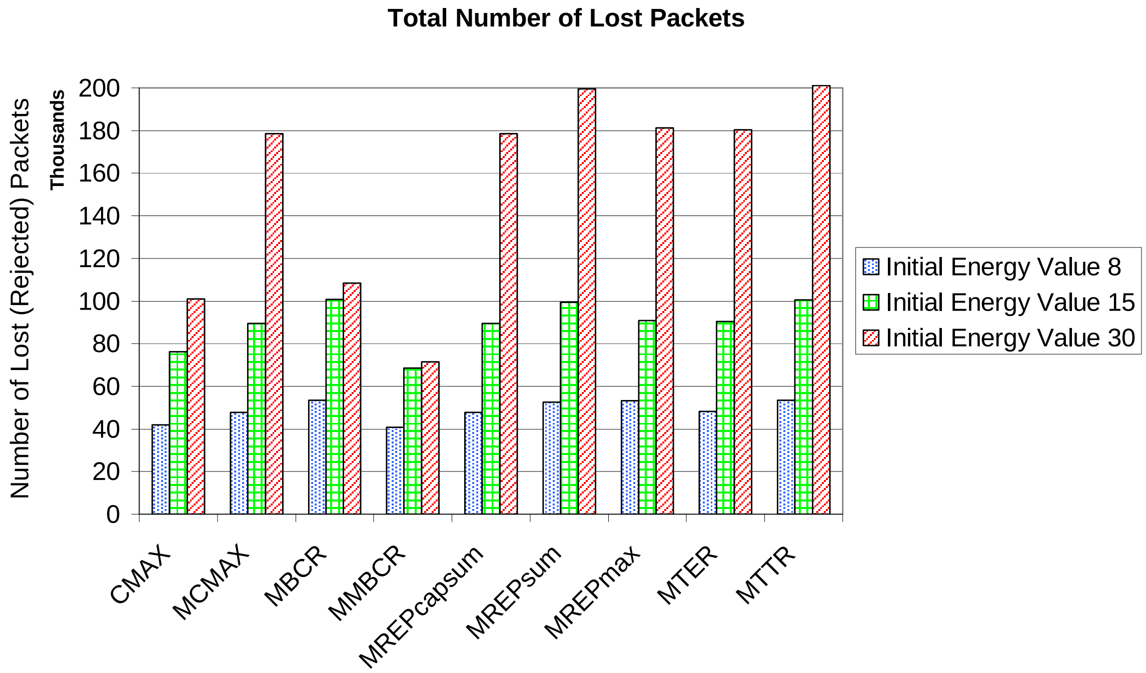


Figure 4.18. Total Number of Lost Packets.

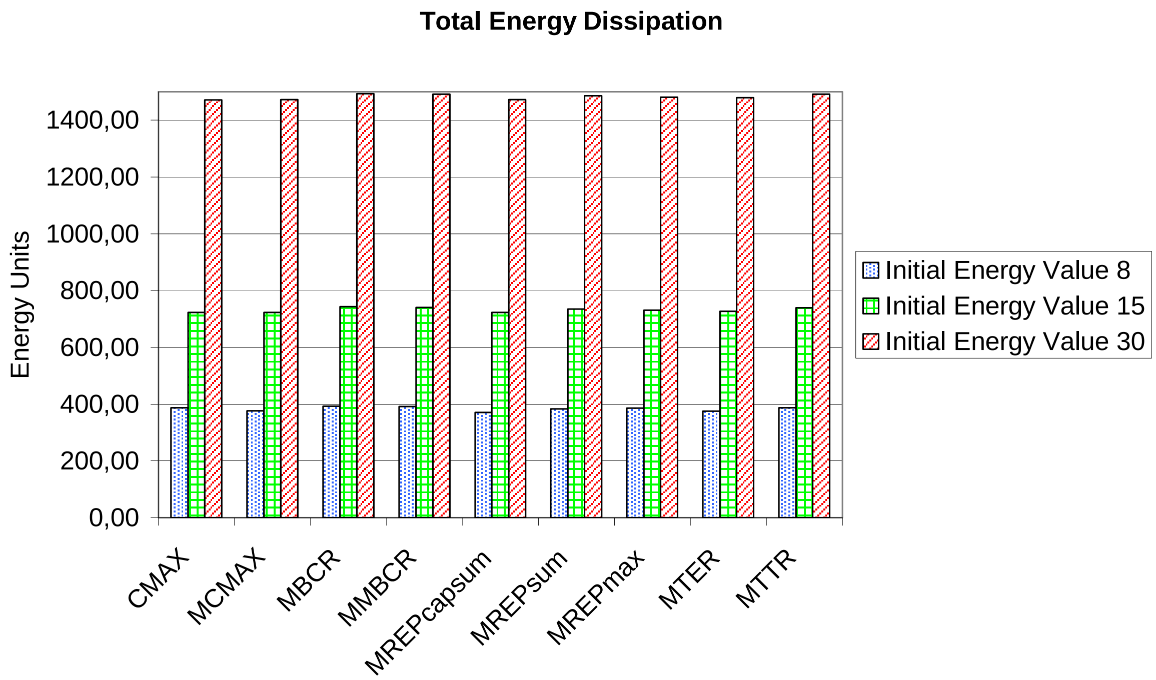


Figure 4.19. Total Energy Dissipated in System during Algorithms Execution.

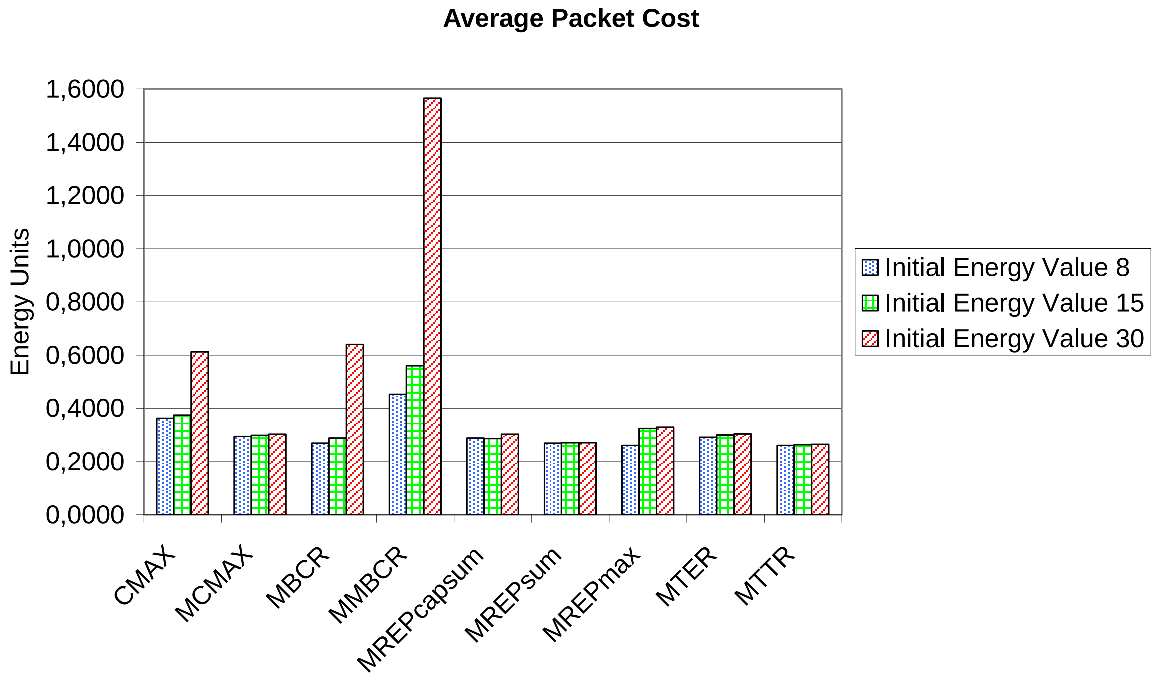


Figure 4.20. Average Cost of Delivering Packet.

	Lifetime	Time of first packet loss	Total throughput	Total num. of lost packets	Total en. diss.	Av. packet cost
Initial energy value of the node – 8 units						
MTER	159	196	1853	48154	375.03	0.2910
MBCR	60	83	2026	53410	392.39	0.2693
MREPcapsum	359	374	1850	47741	371.55	0.2881
MREPsum	245	252	1986	52580	383.27	0.2697
CMAX	151	173	1540	41893	387.28	0.3626
MCMAX	358	367	1846	47732	376.44	0.2932
MMBCR	44	70	1411	40833	390.80	0.4528
MREPmax	125	132	2020	53125	386.28	0.2611
MTTR	106	118	2042	53434	387.29	0.2608
Initial energy value of the node – 15 units						
MTER	293	334	3485	90399	727.30	0.2999
MBCR	84	111	3669	100721	742.65	0.2880
MREPcapsum	712	730	3479	89551	722.62	0.2860
MREPsum	429	438	3769	99477	735.08	0.2712
CMAX	310	332	2840	76283	723.36	0.3733
MCMAX	754	762	3473	89598	722.85	0.2989
MMBCR	65	90	2253	68535	740.84	0.5602
MREPmax	584	587	3353	90840	731.08	0.3244
MTTR	192	223	3852	100613	739.75	0.2638
Initial energy value of the node – 30 units						
MTER	471	509	6983	180298	1480.20	0.3043
MBCR	113	144	3745	108545	1492.70	0.6403
MREPcapsum	1472	1486	6970	178606	1472.40	0.3029
MREPsum	792	804	7602	199460	1486.70	0.2708
CMAX	610	640	3884	101091	1471.00	0.6120
MCMAX	1544	1558	6962	178646	1472.10	0.3031
MMBCR	85	113	2223	71541	1491.30	1.5653
MREPmax	1194	1196	6711	181335	1481.30	0.3286
MTTR	371	410	7728	201016	1491.80	0.2645

Table 4.2. The average simulation results received in the second simulation scenario. Each value is obtained by averaging the results of one hundred simulations.

Appendix 4.A – Theorems

Theorem 1: Let M be a set of nodes and d be another node that is not a member of M : $d \notin M$. The number of packets, denoted by $T(M)$, that can ever be sent by this set of nodes M and delivered to the destination node d is bounded by:

$$T(M) \leq \sum_{j \in M} \left\lfloor \frac{E_j}{e_{jd}^{tx}} \right\rfloor, \tag{4.1}$$

where E_j is the initial energy level of node j and e_{jd}^{tx} is the energy cost of transmitting a packet over the link (j,d).

Proof: Denote the set of all incoming links into destination node d by L_d

$$L_d = \{(j,d) : j \in M\}, \tag{4.2}$$

Any packet delivered to d necessarily traverses some link $(j,d) \in L_d$. Hence the number of packets that arrives at d can be computed as the sum of packets transmitted over all links belonging to set L_d. The number of packets, which was sent over some link (j,d), denoted by $T(j,d)$ using **any** routing strategy cannot exceed the link capacity $W(j,d)$:

$$T(j,d) \leq W(j,d) = \left\lfloor \frac{E_j}{e_{jd}^{tx}} \right\rfloor, \tag{4.3}$$

where $W(j,d)$ is the maximal number of packets that can be transmitted over link (j,d). Therefore, the total throughput (number of packets) delivered to d is bounded by:

$$T(M) = \sum_{j \in M} T(j,d) \leq \sum_{j \in M} W(j,d) = \sum_{j \in M} \left\lfloor \frac{E_j}{e_{jd}^{tx}} \right\rfloor. \tag{4.4}$$

■

There are network applications (e.g. Ad-Hoc Disaster Recovery Networks (DIREN), Sensor Network) where there are a number of possible destinations and the originating node is free to select any one in order to deliver their packets. The theorem stated above can be expanded for this case as follows:

<u>Corollary 1</u>: The number of packets, denoted by $T(M)$, that can be sent by some set of nodes M and delivered to any destination node from some set D is bounded by:

$$T(M) \leq \sum_{j \in M} \left\lfloor \frac{E_j}{\min_{k \in D}(e_{jk}^{tx})} \right\rfloor, \tag{4.5}$$

where E_j is the initial energy level of node j and e_{jd}^{tx} is the energy cost of transmitting a packet over the link *(j,d)*.

<u>Proof</u>: Straightforward. ■

<u>Corollary 2</u>: If the nodes composing the set D, referred to as *sinks*, are not energy-constrained, then the bound (4.5) can be reached by applying the following strategy: a node $j \in M$ that generates a packet, sends it directly, with no help from intermediate nodes, to the closest sink $k \in D$. Packets that cannot be sent directly to their destinations are rejected.

<u>Proof</u>: It is clear that the number of packets transmitted by each sender j to the nearest sink k is equal to the capacity of the directional link *(j,k)*. Moreover, the sent traffic is successfully received by energy-unconstrained sinks. ■

<u>Remark</u>: One can see that the above-mentioned scheme as it is not an *online* routing scheme, because rejects packets that could be delivered to their destinations by indirect path.

Recall algorithm MTTR, suggested on page 40. A node $j \in M$ that generates a packet sends it directly, with no help from intermediate nodes, to the closest sink $k \in D$. If no direct transmission is possible to any of the sinks because energy is low, the packet is sent on a multi-hop route (e.g. minimum transmitted energy route) composed of energy-weak senders, to a node from which direct transmission to a sink is possible. This is an online routing algorithm.

<u>Corollary 3</u>: If the energy consumed for receiving a packet, denoted by e_j^{rx}, is negligible, then the derived bound (4.5) can be reached using the MTTR routing algorithm for any arbitrary generated packet sequence.

Proof: In order to prove the above statement, it is sufficient to show that the total number of packets sent toward the set of sink nodes D equals to the derived bound (4.5). This is because the sent traffic is always successfully received by the sinks. It is clear that, because no energy is consumed by receiving packets, a node that sends a packet does not distinguish between a packet originated by itself and one received from an energy-weak neighbor. Therefore, the total number of packets of both types, denoted by $T(j)$, that a node j can directly send to its preferable sink k in this case is still the capacity of the directional link (j,k). Thus, summation of total throughput sent over all links (j,k), where $j \in M$ and $k \in D$ yields the required equation.

$$\sum_{j \in M} T(j) = T(M) \le \sum_{j \in M} \left\lfloor \frac{E_j}{\min\limits_{k \in D}(e_{jk}^{tx})} \right\rfloor . \tag{4.6}$$

However $T(j)$ satisfies the following condition:

$$T(j) = \left\lfloor \frac{E_j}{\min\limits_{k \in D}(e_{jk}^{tx})} \right\rfloor , \tag{4.7}$$

and therefore

$$\sum_{j \in M} T(j) = T(M) = \sum_{j \in M} \left\lfloor \frac{E_j}{\min\limits_{k \in D}(e_{jk}^{tx})} \right\rfloor . \tag{4.8}$$

■

Theorem 2: Let $G(V,L)$ be a graph. The total throughput (capacity), denoted by $T(G)$, of the network modeled by graph $G(V,L)$ is bounded by:

$$T(G) \le \sum_{j \in V} \left\lfloor \frac{E_j}{\min\limits_{(j,k) \in L}(e_{jk}^{tx})} \right\rfloor . \tag{4.9}$$

where E_j is the initial energy level of node j and e_{jd}^{tx} is the energy cost of transmitting a packet over link (j,k).

Proof: Straightforward. ■

References

[1] J.-H. Chang, L. Tassiulas, "Maximum Lifetime Routing In Wireless Sensor Networks", *Proc. of the 37th Annual Allerton Conference on Communication, Control and Computing*, Sept. 1999.

[2] J.-H. Chang, L. Tassiulas, "Energy Conserving Routing in Wireless Ad-hoc Networks", *IEEE INFOCOM 2000*, March 2000.

[3] K. Kar, M. Kodialam, T.V. Lakshman, L. Tassiulas, "Routing for Network Capacity Maximization in Energy-constrained Ad-hoc Networks", *Proc. of IEEE INFOCOM,* San Francisco pp. 673–681, April 2003.

[4] C.-K. Toh, "Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks", *IEEE Communications Magazine*, June 2001.

[5] Q. Li, J. Aslam, D. Rus, "Online Power-aware Routing in Wireless Ad-hoc Networks", *Proc. of Mobicom 2001*, July 2001.

[6] A. Misra, S. Banerjee, "MRPC: Maximizing Network Lifetime for Reliable Routing in Wireless Environments", *IEEE Wireless Communications and Networking Conference (WCNC)*, Orlando, Florida, volume 2, pp. 800-806, March 2002.

[7] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-efficient Communication Protocols for Wireless Microsensor Networks", *Proc. of the Hawaii International Conference on Systems Science*, Jan. 2000.

[8] M. A. Youssef, M. F.Younis, K. A. Arisha, "A Constrained Shortest-Path Energy-Aware Routing Algorithm for Wireless Sensor Networks", *Wireless Communications and Networking Conference (WCNC'02),* volume 2, pp. 17-21 March 2002.

[9] M. A. Youssef, M. F.Younis, K. A. Arisha, "Energy-Aware Routing in Cluster-Based Sensor Networks", *10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'02)*, Fort Worth, Texas, October 2002.

[10] Carla-Fabiana Chiasserini, Ramesh R. Rao, "Routing Protocols to Maximize Battery Efficiency*", IEEE Milcom 2000*, Los Angeles, USA, October 2000.

[11] Manish Bhardwaj, Timothy Garnett, Anantha P. Chandrakasan, "Upper Bounds on the Lifetime of Sensor Networks", *in Proc. of the International Conference on Communications (ICC'01),* June 2001.

[12] Y. Thomas Hou, Yi Shi, Hanif D. Sherali, "On Rate Allocation Problem for Wireless Sensor Networks with Network Lifetime Requirement", *Technical Report, The Bradley Dept. of ECE*, Virginia Tech, July 2003.

[13] E. Ilker Oyman, Cem Ersoy, "The Effect of Overhead Energy to the Lifetime in Wireless Sensor Networks", *IST Mobile and Wireless Communications Summit*, Lyon, France, June 2004.

[14] Exploratory research: Heterogeneous sensor networks, Intel, Technology Journal: Research & Development at Intel (2004), http://www.intel.com/research/exploratory/heterogeneous.htm.