Proceedings of the 8th IEEE Mediterranean Conference
on Control and Automation (MED 2000)
Rio, Patras, GREECE 17-19 July, 2000

ENERGY CONSIDERATIONS IN MOBILE AD HOC NETWORKS *

Spyros Tragoudas
Electrical and Computer Engineering Department
Southern Illinois Univerity
Carbondale, IL 62901

Abstract The nodes of a mobile ad hoc network run on local energy sources. This paper identifies energy related problems in network management and routing which when appropriately tackled increase the reliability of the network. A new routing method for improved quality of service is presented that takes into consideration the energy life of each node. We also present a methodology for recharging the energy sources of the mobile nodes with minimal impact on the network's reliability.

Key Words. Wireless communication, mobile networks, network reliability, routing protocols

1. INTRODUCTION

Mobile ad hoc networks (MANETs) employ advancements in computer and wireless communication technologies to provide network facilities in situations where no network infrastructure exists [6, 1, 4, 5, 7, 8]. MANETs are very important for military and disaster relief applications. Their advantage is that they can be deployed quickly as no infrastructure is required. The mobility of the nodes and the unreliability of the wireless medium pose challenging tasks in the architecture of a MANET. Another factor that adds to the unreliability of MANETs is due to the a limited amount of energy for the network activities of each local node. We are not aware of any approach that takes into consideration this factor to increase the network's reliability.

This paper assumes that the source that provides energy for the MANET–related activities of node u is independent of the source that provides energy for its mobility. This simplifies the problem formulations and makes it easier for each node to estimate the life of its energy provider. The ideas presented in this paper must be modified to handle nodes that have a common energy source for both activities. Throughout this paper, we are only concerned with the source that supports the MANET activities of mobile node u, and we use the term $el(u)$ to denote its energy life.

At any time unit, each node u in the network can be in one of two states: the on–state or the off–state. While the node is in the on–state it is part of the network. It can relay messages or simply be in a minimal energy consumption sleep mode. While it is in the off–state it is not part of the network. A node switches to the off-state in order to recharge its local energy source. Node u will automatically switch to the off-state when it runs out of energy.

It is not advisable to let the nodes switch to the off-state when they run out of energy because many nodes may switch to the off-state at the same time. This may result to many small disconnected components. In addition, many of those nodes that remain in the on–state will be unable to send messages to nodes that are in the off-state, and the network will be unreliable.

Section 2 describes a methodology for servicing the energy sources of the mobile nodes u with low $el(u)$ values in order to improve the reliability of the network. A scheduling problem is formulated in order to determine when each node will be switched–off to recharge its energy source. Certain objectives must be satisfied to ensure increased network reliability.

Section 3 revisits the routing problem in the presence of $el(u)$ energy bounds on all nodes u on the selected route. It describes a routing approach that guarantees fast data transmission from a node s to a node t subject to the $el(u)$ values. The method ensures that each $el(u)$ on the selected

*Partially supported by NSF grant CCR-0096119

path is large enough to relay the data transmission. The nodes' mobility is also taken into consideration.
Section 4 concludes. It also briefly discusses the efficiency of the presented algorithms.

2. SCHEDULING THE OFF–STATES

2.1 The scheduling problem

Let n be the number of the mobile nodes in the network which is modeled as an undirected graph $G = (V, E)$. The problem of this section ignores the directionality of the network links.
At time t, each node u may locally determine whether it needs to be serviced within the next δ time units. One model that can be applied to estimate $el(u)$ for each node u in a MANET is the path loss model [6] which is discussed in more details in Section 3 later. The value of parameter δ is discussed below.
If $el(u) < \delta$, the node requests from a network management protocol to be in the off–state for r consecutive time slots during the time interval $[t..t + \delta]$. In this case, the node also informs the protocol about its estimated energy life $el(u)$. Let V_k denote the set of nodes that request (at some time t) service on their energy sources within the next δ time units, and $|V_k| = k$.
The protocol discussed in this section will schedule each node in V_k to be in the off-state for r consecutive time units so that certain objectives are satisfied which ensure enhanced network reliability.
The primary objective is to switch the nodes to the off-state so that the network is never disconnected. We call such a schedule a feasible schedule. In particular, we say that a schedule for switching the nodes in V_k to their off-state is feasible if the network which is induced by the nodes in the on–state remains connected every time unit during the interval $[t..t + \delta]$. A feasible schedule clearly enhances the network's reliability for the next δ time units.
Observe that for very small values of δ the scheduling problem deteriorates to immediately switching off each node that needs service and, even though k may not be large, the network will often be disconnected. Thus, δ cannot be too small. However, δ cannot be too large either. It must be small enough to ensure that all nodes' displacement functions are defined or approximated for all δ time units. This will allow for the path–loss model of Section 3 to be applied and estimate $el(u)$ for each $u \in V$. also assume that the value of δ is small enough to avoid turning a node to the off-state more than once during the interval $[t..t + \delta]$.
Obviously, a feasible schedule does not exist if the MANET has an 1-node cut, i.e., a node u whose removal disconnects the MANET. The scheduling of such nodes can be obtained with some preprocessing step that ensures that the network is not decomposed into many connected components.
A secondary objective of the scheduling problem is to control the number of nodes that are turned–off at the same time unit. If many nodes are in the off–state, the network will be underutilized during that time period, and the nodes that remain in the on–state will not be able to perform their scheduled activities. The solutions proposed here for this scheduling problem do not explicitly consider this objective but can be easily modified to take it into consideration.
Before we proceed with providing solutions to the stated scheduling problem, we observe that any node u in V_k that has not been scheduled to switch to the off-state before time $t + el(u)$, it will automatically switch at the latter time and may start servicing its energy source. It is therefore reasonable to assume that the protocol must schedule each node $u \in V_k$ to the off-state for r time units during the time interval $[t..t + el(u)]$. Subsequently, we can assume that $\delta = \max_{u \in V_k}\{el(u)\} + r$.
The scheduling protocol must find a feasible schedule for each node $u \in V_k$ so that it is serviced within the time interval $[t..t+el(u)]$. If there is no feasible schedule, several strategies can be applied. One is to minimize the number of nodes that fail to be serviced within the interval $[t..t+el(u)]$. Under such an objective, we can safely ignore the 1-node cuts in the MANET since their scheduling is invariant to the above objective. Thus, we assume that there are no 1-node cuts in the MANET.

2.1 Proposed solutions

We have shown that following decision problem, we call it the feasibility scheduling (FS) problem, cannot be answered in polynomial time unless P=NP.
The input in the FS problem is an undirected graph $G = (V, E)$, $V_k \subseteq V$, where no $u \in V_k$ is a 1-node cut, $el(u)$, $\forall u \in V_k$, $r \in Z^+$. The goal is to determine whether there exists a feasible schedule for the nodes in V_k, i.e., a schedule where each node in V_k is in the off–state for r

units during the time interval $[t..t + el(u)]$ such that the network remains connected during each of the $\delta = \max_u\{el(u)\} + r$ time units.

This result implies that it is very unlikely that a polynomial time algorithm exists for FS problem, and we have to rely on heuristics. The presented heuristics utilize the following lemma.

Lemma 1 *A feasible schedule, if one exists, can be obtained even if we insist that $u \in V_k$ is in the off-state with the same set of nodes in V_k for all r time units.*

The heuristics tackle an optimization version of the FS problem where the goal is to find the minimum possible number k' of nodes $u \in V_k$ that are not scheduled before their deadline $el(u)$ while insisting that the remaining $k - k'$ nodes are scheduled as maximal sets, i.e., sets of elements from nodes in V_k that cannot be augmented because the network will be disconnected. Lemma 1 allows us to assume that r=1 when developing heuristics for the FS optimization problem.

In the remaining of this section, we outline heuristics for the special case when all $el(u)$ are equal to $\delta - r$, $\delta \in Z^+$. The heuristics need to be modified in order to tackle the more general case.

When all $el(u)$ are the same, the goal in the FS optimization problem is to obtain the minimum time δ_{OPT} that guarantees a feasible schedule (consisting of maximal sets from elements in V_k). In this case, a solution close to δ_{OPT} is very likely to result to a feasible schedule, if one exists.

We model the uniform FS optimization problem as the optimization version of the much studied set covering problem [3]. For completeness, we define the set covering problem in the following.

Set covering problem (optimization version): The input is a set V_k of elements and a collection P of p sets s_i of elements from V_k. The goal is to find the minimum number of sets from P so that every element of V_k is covered, i.e., belongs to at least one of the selected sets from P.

The FS optimization problem can be reduced to the set covering optimization problem by assuming an enumeration of all maximal sets of V_k, i.e., sets of nodes in V_k whose removal does not disconnect graph $G = (V, E)$. Observe that when all $el(u)$ are equal the exact ordering (in consecutive time units) of the selected maximal sets of elements in V_k is invariant. Thus, the cardinality of a minimum set cover is equal to δ_{OPT}.

The following greedy heuristic has been shown to perform well for the set covering optimization problem.

```
Set_Cover (V_k, P)
U = V_k
C = ∅
i = 0
while U ≠ ∅
   select a set s_i from P that maximizes |s_i ∩ U|
   U = U \ s_i
   C = C ∪ {s_i}
   i = i + 1
end Set_Cover
```

It has been shown that Set_Cover never selects more than $(\ln s_{max} + 1) \cdot \text{OPT}$ sets s_i, where OPT is the minimum number of sets needed to cover the elements of V_k, and s_{max} is the maximum cardinality of any set in V_k [3]. Thus, we have:

Theorem 1 Set_Cover never requires more than $\delta_{GSC} = (\ln s_{max} + 1) \cdot \delta\text{OPT}$ time units for servicing all the nodes in V_k, where δOPT is the duration of the shortest possible schedule that can satisfy all the nodes in V_k.

Set_Cover is expected to return a feasible schedule, i.e., a schedule where $\delta_{GSC} \leq \delta$, if $\delta\text{OPT} \ll \delta$. The obvious disadvantage of heuristic Set_Cover is that it requires an enumeration of the maximal subsets of V_k that do not disconnect G, and may not have polynomial time complexity. Nevertheless, it can be used to evaluate the quality of faster heuristics. It also justifies the following polynomial time algorithm which generates the cover C without considering set P.

```
Greedy_Set_Cover (G, V_k)
U = V_k
C = ∅
i = 0
while U ≠ ∅
   call form_set (s_i)
   U = U \ {s_i}
   C = C ∪ {s_i}
   i = i + 1
end Greedy_Set_Cover
```

The goal in procedure form_set(s_i) is to generate a set s_i that covers as many new unscheduled nodes from U as possible. The larger the cardinality of s_i, the closer the performance of

Greedy_Set_Cover() to that of Set_Cover() is. In the following we outline a fast implementation for procedure form_set(s_i).

procedure form_set (s)
select node $u \in U$
$s = \{u\}$
for each $v_i \in U$ do
  $s = s \cup \{u\}$ provided that the augmentation of s does not disconnect G
  end Greedy_Set_Cover

The time complexity of such an implementation for Greedy_Set_Cover is $O(k^2 \cdot m)$, where $k = |V_k|$ and m is the number of links in the network.

3. ROUTING THROUGH NODES WITH BOUNDED ENERGY

3.1 The routing problem

This section presents a new routing method for improved quality of service in mobile ad hoc networks. The route selection takes into consideration an estimate for the duration of the transmission, the energy life $el(u)$ of each node in the network, and an estimate of the energy consumption to transmit and receive a signal along each link. The routing approach guarantees the shortest feasible data transmission that can be supported by the energy sources for the nodes on the selected route.

Many routing protocols have been presented recently for MANETs. See [1,2], among recent approaches. Existing approaches compute the route as a shortest path. The main focus is on scalability, and less effort has been devoted in increasing the reliability of the transmission.

Let $N = (V, E, c, l, k, d, b)$ be an n-node, m-link network, where $G = (V, E)$ is a directed graph, c is the capacity function that assigns an integer capacity on each directed link, l is the length function that assigns distances on links at any time unit based on the velocities of the endpoints, k is a function that assigns an integer upper bound for each link above which the link is disconnected, d is the lead delay function that determines the average wait time needed to transmit the header of a packet from the source of a link to its sink, and el is the function that estimates the energy life of each node in the MANET. Let also s, t be two network nodes, the source and the target, and an integer σ equal to the amount of data to be sent from node s to node t. The goal is to find a single path p to transmit the σ units of data from the source s to the sink t.

Let the lead time $d(p)$ along a path p be the sum of the delays of its links. This is equal to the amount of time needed to send the lead packet from the source to the sink along p. The path capacity (or bottleneck) $c(p)$ is defined as the minimum link capacity on path p.

The total transmission time $T(\sigma, p)$ required to send σ units of data along path p is

$$T(\sigma, p) = d(p) + \frac{\sigma}{c(p)}.$$

Quantity $\frac{\sigma}{c(p)}$ denotes the duration of the data transmission along p citereliable. Observe that $T(\sigma, p)$ is not impacted by the nodes' mobility. However, the following show that the energy consumption of the nodes u on the path p for relaying the σ units of data is impacted by mobility. According to the path–loss model [6], each transmitter u on path p consumes approximately

$$\pi \cdot l(u, v)^4$$

power to send a single message to receiver v along link (u, v), where π is a predetection constant threshold (in mW) for each node in the network. This is a simplified version of the path–loss model for MANETs where we can safely assume that all antenna heights are different. Thus, the power does not depend on the antenna heights. However, the power is clearly impacted by the mobility of nodes u and v since $l(u, v)$ may be affected by the mobility of nodes u and v.

The power for receiving a signal at node v is a constant ρ for all the nodes in the network, and therefore is not impacted by mobility. The following theorem states the total energy required to transmit σ units of data through each node on path p.

Theorem 2 The total energy required to send σ units of data along a path p is distributed along the nodes on path p as follows:

$\sigma \cdot \pi \cdot l(s, v)$ amount of energy for the source s that transmits the data along the directed link $(s, v) \in p$

$\sigma \cdot \pi \cdot l(u, v) + \rho$ amount of energy for any relay node $u \neq s, t$ that transmits them along the directed link $(u, v) \in p$

ρ amount of energy for the target t.

The amount of energy consumed by each node on the path (except the target t) is affected by mobility due to the changes in the distances on the links on the path. However, the energy is independent of the duration of the transmission $\frac{\sigma}{c(p)}$.

3.2 The fastest feasible transmission

This section describes a dynamic programming algorithm for finding a path p between s and t that minimizes quantity $T(\sigma, p)$ subject to adequate energy on each node u on the path. In this section we assume that the nodes are stationary, i.e., the $l(u, v)$ values do not change over time. The following section generalizes the approach to handle node mobility.

Given path p one can compute the required energy for each node u on p using Theorem 2. Observe that the energy for each node u only depends on which incident link was used for the data transmission. This allows us to preprocess the network as follows: We assume that $el(t) \geq \rho$, otherwise no feasible route exists.

Remove all outgoing edges $l(s, v)$ from source s for which $\sigma \cdot \pi \cdot l(s, v)^4 > el(s)$.

Remove any relay node $u \neq s, t$ for which $el(u) < \rho$.

For any remaining relay node $u \neq s, t$ in the network, remove all outgoing edges $l(u, v)$ for which $el(u) - \rho < \sigma \cdot \pi \cdot l(u, v)^4$.

Let $G = (V, E)$ be the induced network. Any path p connecting nodes s and t in G is a feasible path, i.e., there is enough energy at the local nodes for relaying the data. The following describe an algorithm for calculating the (s, t) path that guarantees the smallest $T(\sigma, p)$ value.

We have observed that calculating such a path requires a method different than a simple shortest path calculation. All existing algorithms for the well known shortest path problem rely on a simple principle of optimality that states that any (s', t') subpath of a shortest path p (with nodes s', t' on path p) must itself be a shortest path connecting s' and t'.

Unfortunately, a fastest $T(\sigma, p)$ path p between nodes s and t does not necessarily satisfy the property that every subpath p' on it is itself a fastest subpath for transmitting the same amount of data between its endpoints. The reason is that $c(p')$ may be larger than $c(p)$.

This problem could be easily bypassed if we knew the value of the capacity $c(p)$ of the optimal path p. Let N_{c_i} be the network obtained from N by removing all links with capacity smaller c_i. There are at most $c_m \leq m$ different capacities in the network and thus c_m networks N_{c_i}. If $c(p)$ where equal to c_i, $1 \leq i \leq m$, then the shortest path of network N_{c_i} (considering the $d(u, v)$ link values) guarantees the fastest $T(\sigma, p)$ transmission from s to t. This observation suggests the following simple algorithm whose complexity is $O(c_m)$ times the complexity of a shortest path algorithm.

Fastest_transmission (G, s, t, σ)

Compute a shortest path p_{c_i} for each network $N_{c_i}, 1 \leq c_i \leq c_m$, using the $d(u, v)$ values on the links of N_{c_i}.

The selected path p is the path p_{c_i} for which the quantity $d(p_{c_i}) + \frac{\sigma}{c(p_{c_i})}$ is minimized.

3.3 Handling node mobility

Algorithm Fastest_transmission is modified to handle mobility. The modified algorithm assumes node velocities that are translated to a displacement function for each node. The measure of mobility is actually the displacement of the nodes in each coordinate direction. When nodes u, v are moving, the $l(u, v)$ are changing. When $l(u, v) > k(u, v)$ a link the link is deleted, and vice versa. Furthermore, the change in $l(u, v)$ impacts the energy at $el(u)$.

Let $l(u, v)^t$ denote the distance between nodes u and v at time t. We assume that we able to compute efficiently the distance $l(u, v)^t$ from the displacement functions for nodes u and v.

If we could have guessed that the data transmission time $\frac{\sigma}{c(p)}$ along a fastest path p has value t_g, then Theorem 2 could be applied for each of the t_g time units to remove links in the network that cause infeasibility. We could then apply algorithm Fastest_transmission to find an optimal path p that guarantees the fastest total transmission time $T(\sigma, p)$. The optimal transmission paths can be partitioned into up to c_m equivalence classes according to their data transmission time $t_r = T(\sigma, p) - d(p)$. One of those transmission times t_r must be equal to the guessed data transmission time t_g.

Let us implement algorithm Fastest_transmission has been implemented so that if more than one network N_{c_i} contain optimal paths, it return all the respective data transmission times t_r. Observe that both t_g as well as the data transmission time of an optimal path are in the range $[1..\sigma]$.

Let us initially guess that $t_g = 1$. We remove appropriate links as indicated above and then we run algorithm Fastest_transmission that returns all possible t_r values.

We repeat the above process with $t_g = t_g + 1$ unless $t_g = t_r$, at which point we have found an optimal path p. Since the data transmission time of any optimal is in the range $[1..\sigma]$, the process is ensured to terminate within at most σ iterations. The complexity of the described algorithm is $O(\sigma \cdot (\sigma \cdot m + c_m \cdot (m + n \cdot \log n)))$. The multiplicative factor σ is due to the $O(\sigma)$ iterations, one per t_g value. Observe that the first term $\sigma \cdot m$ in the second factor $O(\sigma \cdot m + c_m \cdot (m + n \cdot \log n))$ must always exist in the time complexity by the problem definition. This happens because we cannot avoid calculating the $l(u, v)$ distances at each of the at most $\frac{\sigma}{c(p)} = O(\sigma)$ time units using the displacement functions of the nodes in the network. We must be aware of the network topology each one of the above time units. Thus, the above is not a pseudopolynomial time algorithm.

4. CONCLUDING REMARKS

Two problems that relate to the reliability of mobile ad hoc networks have been studied in this paper. Both problems indicate that the reliability of the network may be impacted by the limited amount of energy at each node in the network. We have presented solutions for both problems so that the reliability of the network is deteriorating as much as possible. The presented algorithms need to be implemented in a distributed manner. Algorithms Greedy_Set_Cover() for the first problem and Fastest_transmission for the second problem have been implemented and run on the network topology of the ISCAS'85 benchmarks which are widely used in CAD for VLSI. The largest of those benchmarks has more than 7,500 links and over 3,500 nodes. Despite the large network sizes, the algorithm terminated within a few minutes when executed on a Ultra 10 Sun workstation. It is important that fast algorithms are developed for MANETs in order to support scalability.

References

[1] Chen S., Nahrstedt K. Distributed QoS Routing in Ad Hoc Networks, , IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, pp. 1488-1505, August 1999.

[2] Chen W., Jain N., Singh S. ANMP: Ad Hoc Network Management Protocol, IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, pp. 1506-1531, August 1999.

[3] Cormen T.H., Leiserson C.E., Rivest R.L. Introduction to algorithms, McGraw-Hill, 1990.

[4] Iawata A., Chiang C.-C., Pei G., Gerla M., Chen T.-W. Scalable Routing Strategies for Mobile Ad Hoc Networks, , IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, pp. 1369-1379, August 1999.

[5] Pearlman M.R., Haas Z.J. Optimal configuration for the zone routing protocol, IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, pp. 1395-1414, August 1999.

[6] Radoplu V., Meng T.H. Minimum Energy Mobile Ad Hoc Networks, , IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, pp. 1333-1344, August 1999.

[7] Ramanathan R., Steenstrup M. Hierarchically Organized, Multi- Hop Mobile Networks for Quality of Service Support, Mobile Networks and Applications, vol. 3, no. 1, pp. 101-119, 1998.

[8] Sivakumar R., Sinha P., Bhagravan V. CEDAR: A Core Extraction Distributed Ad Hoc Routing Algorithm, , IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, pp. 1454- 1465, August 1999.

[9] Tragoudas S. The most reliable data path transmission, 1999 International Performance, Computing, and Communications Conference, pp. 15–19, Feb. 10–12, 1999, Scottdale, AZ, USA.