

Fast Approximate Algorithms for Maximum Lifetime Routing in Wireless Ad-hoc Networks *

Jae-Hwan Chang and Leandros Tassiulas

Department of Electrical & Computer Engineering
and Institute for Systems Research
University of Maryland, College Park MD 20742, USA,
{jhchang,leandros}@isr.umd.edu

Abstract. In wireless ad-hoc networks, routing and power consumption are intrinsically connected since the transmit power level is adjusted depending on the location of the next hop node. Our objective is to route the traffic such that the system lifetime is maximized. In the previous works, we have formulated the problem and proposed heuristic algorithms. In this paper, fast approximate algorithms are proposed both in single commodity case and in the multicommodity case along with the analysis which enables us to control the trade-off between the performance bound and the running time. The problem is then extended to have an additional constraint on the delay.

1 Introduction

Consider a group of randomly distributed wireless static nodes as in Fig. 1, where each node has a limited battery energy used mainly for the transmission of data. Assume that at each node some type of information is generated, which needs to be delivered to some nodes designated as gateway nodes. An example may be a wireless sensor network where the sensors gather acoustic, magnetic, or seismic information and send the information to its gateway node which has more processing power for further processing of the information or has larger transmission range for the delivery of the information to a possibly larger network for retrieval by a remote user. The wireless nodes are assumed to have the capability of packet forwarding, i.e., relaying an incoming packet to one of its neighboring nodes, and the transmit power level can be adjusted to a level appropriate for successful reception if the receiver is within the transmission range. Upon or before a new arrival of information either generated at the node itself or forwarded from the other nodes, routing decision has to be made so that the node knows which of its neighboring nodes to send its data to. Note that the routing decision and the transmission energy level selection are intrinsically

* Prepared through collaborative participation in the Advanced Telecommunications/Information Distribution Research Program (ATIRP) Consortium sponsored by the U.S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0002.

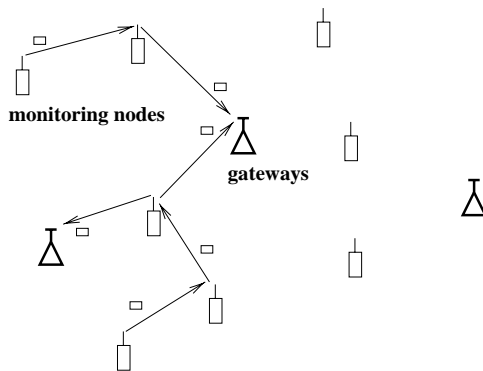


Fig. 1. A multi-hop wireless ad-hoc network is depicted where the information generated at the monitoring nodes are delivered to the gateway nodes.

connected in this power-controlled ad-hoc network since the power level will be adjusted depending on the location of the next hop node.

Most of the previous works on routing in wireless ad-hoc networks deal with the problem of finding and maintaining correct routes to the destination during mobility and changing topology [2,6,12]. In [2,6], the authors presented a simply implementable algorithm which guarantees strong connectivity and assumes limited node range. Shortest-path algorithm is used in this strongly connected backbone network. In [12], the authors developed a dynamic routing algorithm for establishing and maintaining connection-oriented sessions which uses the idea of predictive re-routing to cope with the unpredictable topology changes. Some other routing algorithms in mobile wireless networks can be found in [10,13,14,15], which, as the majority of routing protocols in mobile ad-hoc networks do, use shortest-path routing where the number of hops is the path length.

The problem of minimum energy routing has been addressed before [2,6,7], [8,11,16,17,18]. The approach in these works was to minimize the consumed energy to reach the destination. In [3], knowing that the approach doesn't provide an optimization in the network level, we have formulated the maximum system lifetime routing problem as a linear programming problem, and proposed heuristic algorithms which showed close-to-optimal performance in the simulation results. In this paper, we will propose fast approximate algorithms with the controllable performance bound and the running time based on the work in [9] with some modifications necessary for our problem.

In our study the topology of the network is static and the routing accounts to finding the traffic splits that balance optimally the energy consumption. Hence the results are applicable to networks which are either static, like the sensor networks we mentioned earlier, or their topology changes slowly enough such that there is enough time for optimally balancing the traffic in the periods between successive topology changes.

This paper is organized as follows: In Section 2, the problem is formulated. In Section 3, fast approximate algorithms are proposed in the single commodity case and in the multicommodity case. In Section 4, the problem is extended to the case where there is an additional constraint on the delay.

2 Flows, Optimal Energy Consumption and Transmission Power Levels

In this section, the problem formulation given in our previous work[3] is repeated for the sake of completeness of the presentation.

The wireless ad-hoc network in consideration is modeled as a directed graph $G(N, A)$ where N is the set of all nodes and A is the set of all directed links (i, j) where $i, j \in N$. Let S_i be the set of all nodes that can be reached by node i with a certain power level in its dynamic range. We assume that link (i, j) exists if and only if $j \in S_i$. Let each node i have the initial battery energy E_i , and let $Q_i^{(c)}$ be the rate at which information is generated at node i belonging to commodity $c \in C$, where C is the set of all commodities. We are given, for each commodity c , an origin node $o^{(c)}$ where the information is generated, and a destination node $d^{(c)}$. Assume that the transmission energy required for node i to transmit an information unit to its neighboring node j is e_{ij} , and the rate at which information of commodity c is transmitted from node i to node j is called the flow $q_{ij}^{(c)}$.

The lifetime of node i under a given flow $\mathbf{q} = \{q_{ij}^{(c)}\}$ is given by

$$T_i(\mathbf{q}) = \frac{E_i}{\sum_{j \in S_i} e_{ij} \sum_{c \in C} q_{ij}^{(c)}}. \quad (1)$$

Now, let us define the *system lifetime* under flow \mathbf{q} as the minimum lifetime over all nodes, i.e.,

$$T_{sys}(\mathbf{q}) = \min_{i \in N} T_i(\mathbf{q}). \quad (2)$$

Our goal is to find the flow that maximizes the system lifetime under the flow conservation condition drawn in Fig.2, and the problem can be written as follows:

$$\begin{aligned} & \text{Maximize} && T_{sys}(\mathbf{q}) \\ & \text{s.t.} && q_{ij}^{(c)} \geq 0, && \forall i \in N, \forall j \in S_i, \forall c \in C, \\ & && \sum_{j: i \in S_j} q_{ji}^{(c)} + Q_i^{(c)} = \sum_{k \in S_i} q_{ik}^{(c)}, && \forall c \in C, \forall i \in N - \{d^{(c)}\}. \end{aligned} \quad (3)$$

The problem of maximizing the system lifetime, given the information generation rates $Q_i^{(c)}$ at the origin node $o^{(c)}$ for each commodity c , can be rewritten

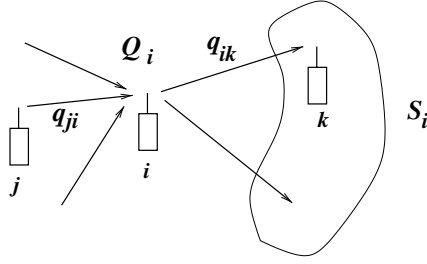


Fig. 2. The conservation of flow condition at node i is illustrated.

as the following linear programming problem:

$$\begin{aligned}
 & \text{Maximize} && T \\
 & \text{s.t.} && \hat{q}_{ij}^{(c)} \geq 0, && \forall i \in N, \forall j \in S_i, \forall c \in C, \\
 & && \sum_{j \in S_i} e_{ij} \sum_{c \in C} \hat{q}_{ij}^{(c)} \leq E_i, && \forall i \in N, \\
 & && \sum_{j: i \in S_j} \hat{q}_{ji}^{(c)} + TQ_i^{(c)} = \sum_{k \in S_i} \hat{q}_{ik}^{(c)}, && \forall c \in C, \forall i \in N - \{d^{(c)}\},
 \end{aligned} \tag{4}$$

where $\hat{q}_{ij}^{(c)} = Tq_{ij}^{(c)}$ is the amount of information of commodity c transmitted from node i to node j until time T . Note that when the information generation rate is fixed, the objective of maximizing the system lifetime is equivalent to maximizing the amount of data transfer.

The linear program given above can be viewed as a variation of the conventional maximum flow problem with node capacities[5]. If the transmitted power level at each node is fixed regardless of its next hop node, i.e., if there is no power control, then

$$e_{ij} = e_i, \quad \forall j \in S_i, \tag{5}$$

and the problem is equivalent to the maximum flow problem with node capacities given by

$$\sum_{j \in S_i} \sum_{c \in C} \hat{q}_{ij}^{(c)} \leq E_i/e_i, \quad \forall i \in N. \tag{6}$$

When the capacity of a node is a fixed quantity as in (6) then the problem can be converted to a link capacity version by replacing the node with two nodes and a link having the same capacity[4], and the max-flow-min-cut theorem[5] can be used. However, in our problem, unlike the above, the amount of resource (or energy in this case) which a unit flow consumes depends on the energy expenditure to the next hop node. Therefore, it is not trivial to find the min-cut nodes, and even if they were found the traffic split at the nodes must also be identified.

3 Fast Approximate Algorithms

In this section, fast approximate algorithms will be proposed in single commodity case and in multicommodity case along with the analysis which enables us to control the performance and the complexity of the algorithms.

3.1 Single Commodity Case

Let o and d denote the origin node and the destination node, respectively. Let P be the set of all paths between o and d , and let f_i be the value of the flow on the i th path in P . The problem in (4), in the single commodity case, can be rewritten as follows:

$$\begin{aligned} & \text{Maximize} && \sum_{i \in P} f_i \\ & \text{s.t.} && f_i \geq 0, \quad \forall i \in P, \\ & && \sum_{j \in \mathcal{P}_i} e_{l(i,j)} f_j \leq E_i, \quad \forall i \in N - \{d\}, \end{aligned} \quad (7)$$

where \mathcal{P}_i is the set of all paths that passes node i , and $l(i, j)$ is the link on path j that stems from node i .

The problem above is a linear program and its dual problem is given by

$$\begin{aligned} & \text{Minimize} && \sum_{i \in N} \lambda_i E_i \\ & \text{s.t.} && \lambda_i \geq 0, \quad \forall i \in N, \\ & && \sum_{j \in \mathcal{N}_i} \lambda_j e_{l(j,i)} \geq 1, \quad \forall i \in P, \end{aligned} \quad (8)$$

where \mathcal{N}_i is the set of all nodes on path i except the destination node d .

Physical interpretation of the dual problem is the assignment of node weights λ_i such that $\sum_{i \in N} \lambda_i E_i$ is minimized with the constraint that the length of the shortest length path is at least one, where the length of a path is defined as the weighted sum of energy expenditure per unit flow over each link in the path by the weight of its head node.

The proposed algorithm is described in the following.

Let $J(\lambda(x))$ be defined as the objective function at the x th iteration, i.e.,

$$J(\lambda(x)) = \sum_{i \in N} \lambda_i(x) E_i,$$

and let $\alpha(\lambda(x))$ be defined as the length of the shortest path $sp(x)$ at the x th iteration, i.e.,

$$\begin{aligned} \alpha(\lambda(x)) &= \min_{j \in P} \sum_{i \in \mathcal{N}_j} \lambda_i(x) e_{l(i,j)} \\ &= \sum_{i \in \mathcal{N}_{sp(x)}} \lambda_i(x) e_{l(i, sp(x))}. \end{aligned}$$

The problem is equivalent to minimizing $\frac{J(\lambda(x))}{\alpha(\lambda(x))}$, and let β be this minimum, i.e.,

$$\beta = \min_{\lambda(x)} \frac{J(\lambda(x))}{\alpha(\lambda(x))}.$$

Let each node i have initial weight given by

$$\lambda_i(0) = \frac{\delta}{E_i}, \quad (9)$$

where δ is a constant which will be determined later. At the x th iteration, an amount of $\Delta f(x)$ of flow is augmented on the shortest path $sp(x-1)$, where $\Delta f(x)$ is given by

$$\Delta f(x) = \min_{i \in \mathcal{N}_{sp(x-1)}} \frac{E_i}{e_{l(i, sp(x-1))}}, \quad (10)$$

which is the capacity of the minimum capacity link on the shortest path $sp(x-1)$. If we denote the total flow assigned until the x th iteration by $f(x)$ then $f(0) = 0$ and $f(x) = f(x-1) + \Delta f(x)$. After the augmentation, the node weights on the shortest path are updated by

$$\lambda_i(x) = \lambda_i(x-1) \{1 + \epsilon \Delta f(x) e_{l(i, sp(x-1))} / E_i\}, \quad (11)$$

for all $i \in \mathcal{N}_{sp(x-1)}$, where ϵ is a constant which will be determined later. Note that $\lambda_i(x) = \lambda_i(x-1)$ for all the other nodes that are not on the shortest path. Using the updated $\lambda_i(x)$, shortest path $sp(x)$ is calculated. The shortest path calculation and the flow augmentation are repeated until iteration t where $J(\lambda(t)) \geq 1$ for the first time.

In the following, we analyze the algorithm's performance and its running time.

For all $x \geq 1$,

$$\begin{aligned} J(\lambda(x)) &= \sum_{i \in N} \lambda_i(x-1) E_i + \epsilon \Delta f(x) \sum_{i \in \mathcal{N}_{sp(x-1)}} \lambda_i(x-1) e_{l(i, sp(x-1))} \\ &= J(\lambda(x-1)) + \epsilon \{f(x) - f(x-1)\} \alpha(\lambda(x-1)). \end{aligned}$$

Since

$$\beta \leq \frac{J(\lambda(x-1))}{\alpha(\lambda(x-1))},$$

it follows that

$$\begin{aligned} J(\lambda(x)) &\leq J(\lambda(x-1)) \left[1 + \frac{\epsilon}{\beta} \{f(x) - f(x-1)\}\right] \\ &\leq J(\lambda(x-1)) \exp\left(\frac{\epsilon}{\beta} \{f(x) - f(x-1)\}\right), \end{aligned}$$

which implies that

$$J(\lambda(x)) \leq n\delta \exp\left(\frac{\epsilon}{\beta} f(x)\right).$$

By our stopping condition,

$$1 \leq J(\lambda(t)) \leq n\delta \exp\left(\frac{\epsilon}{\beta} f(t)\right).$$

Therefore,

$$\frac{\beta}{f(t)} \leq \frac{\epsilon}{\ln(n\delta)^{-1}}. \quad (12)$$

There is a feasible flow of value $\frac{f(t)}{\log_{1+\epsilon} \frac{1+\epsilon}{\delta}}$, which can be shown as follows. Consider node i . For every E_i consumed, the weight increases by a factor of at least $1 + \epsilon$. Before the algorithm stops, $J(\lambda(t-1)) = \sum_{i \in N} \lambda_i(t-1)E_i < 1$, which means that $\lambda_i(t-1) < 1/E_i$ for each i . At each iteration the node weight is increased by a factor of at most $(1 + \epsilon)$, so $\lambda_i(t) < (1 + \epsilon)/E_i$. Initially, the node weight was δ/E_i , and E_i may have been consumed at most t times. Therefore,

$$\frac{\delta}{E_i} (1 + \epsilon)^t < \frac{(1 + \epsilon)}{E_i},$$

and the total consumed energy at node i is at most

$$E_i \log_{1+\epsilon} \frac{1 + \epsilon}{\delta}.$$

The final flow $f(t)$ may have violated the energy constraint, and can be made feasible by dividing it by $\log_{1+\epsilon} \frac{1+\epsilon}{\delta}$. Note that the resulting flow from this algorithm represents the amount of possible information transfer. To obtain the system lifetime, it should be divided by the information generation rate.

Now, the ratio between the optimum dual and the primal solutions is given by

$$\gamma = \frac{\beta}{f(t)} \log_{1+\epsilon} \frac{1 + \epsilon}{\delta}.$$

Substituting this to (12) gives

$$\begin{aligned} \gamma &\leq \frac{\epsilon \log_{1+\epsilon} \frac{1+\epsilon}{\delta}}{\ln(n\delta)^{-1}} \\ &= \frac{\epsilon}{\ln(1 + \epsilon)} \frac{\ln \frac{1+\epsilon}{\delta}}{\ln(n\delta)^{-1}}. \end{aligned}$$

The ratio $\frac{\ln \frac{1+\epsilon}{\delta}}{\ln(n\delta)^{-1}}$ equals $(1 - \epsilon)^{-1}$ when

$$\delta = (1 + \epsilon) \{(1 + \epsilon)n\}^{-1/\epsilon}. \quad (13)$$

Hence with this choice of δ we have

$$\begin{aligned} \gamma &\leq \frac{\epsilon}{(1 - \epsilon) \ln(1 + \epsilon)} \\ &\leq \frac{\epsilon}{(1 - \epsilon)(\epsilon - \epsilon^2/2)} \\ &\leq \frac{1}{(1 - \epsilon)^2}. \end{aligned}$$

Therefore, ϵ is chosen to be

$$\epsilon = 1 - \frac{1}{\sqrt{\gamma}}, \quad (14)$$

in order to guarantee the γ -approximation.

The total number of iterations in which a node is the head of the minimum capacity link on the path chosen in that iteration is at most

$$\lceil \log_{1+\epsilon} \frac{1}{\delta} \rceil = \lceil \frac{1}{\epsilon} - 1 + \frac{1}{\epsilon} \log_{1+\epsilon} n \rceil.$$

Using the fact that there are n nodes we get the following running time. The running time of the algorithm that computes a $(1 - \epsilon)^{-2}$ -approximation to the optimal solution is at most

$$n \lceil \frac{1}{\epsilon} - 1 + \frac{1}{\epsilon} \log_{1+\epsilon} n \rceil T_{sp},$$

which is $O(n(\frac{1}{\epsilon} \log_{1+\epsilon} n) T_{sp})$, where T_{sp} is the time required to compute the shortest path in a graph with non-negative link lengths.

3.2 Multicommodity Case

Assume that each commodity has a single origin and destination pair. Note that the case of having multiple destinations where any one of them needs to be reached can be obtained by inserting an imaginary node connecting all destinations, and the case of having multiple origins can be obtained by treating each as a separate commodity. Let $Q^{(c)}$ be the information generation rate of commodity c at its origin node $o^{(c)}$. Let \hat{P} be the set of all path clusters where a path cluster consists of one path for each commodity. Let f_i be such that the flow value of commodity c on the path cluster i is $Q^{(c)} f_i$. Note that we are maximizing the flow while keeping the ratios among the information generation rates of all commodities. The problem in (4) can be rewritten as follows:

$$\begin{aligned} & \text{Maximize} && \sum_{i \in \hat{P}} f_i \\ & \text{s.t.} && f_i \geq 0, \quad \forall i \in \hat{P}, \\ & && \sum_{c \in C} \sum_{j \in \mathcal{P}_i^{(c)}} e_{l^{(c)}(i,j)} Q^{(c)} f_j \leq E_i, \quad \forall i \in N, \end{aligned} \quad (15)$$

where $\mathcal{P}_i^{(c)}$ is the set of all path clusters whose path for commodity c passes node i , and $l^{(c)}(i,j)$ is the link that commodity c of path cluster j visits right after visiting node i .

The dual problem is given by

$$\begin{aligned} & \text{Minimize} && \sum_{i \in N} \lambda_i E_i \\ & \text{s.t.} && \lambda_i \geq 0, \quad \forall i \in N, \\ & && \sum_{c \in C} Q^{(c)} \sum_{j \in \mathcal{N}_i^{(c)}} \lambda_j e_{l^{(c)}(j,i)} \geq 1, \quad \forall i \in \hat{P}, \end{aligned} \quad (16)$$

where $\mathcal{N}_i^{(c)}$ is the set of all nodes on the path for commodity c of path cluster i .

The algorithm is described in the following.

Let $J(\lambda(x))$ be defined as

$$J(\lambda(x)) = \sum_{i \in N} \lambda_i(x) E_i,$$

and let $\alpha(\lambda(x))$ be defined as

$$\alpha(\lambda(x)) = \frac{1}{m} \sum_{c \in C} Q^{(c)} \sum_{i \in \mathcal{N}_{sp^{(c)}(x)}^{(c)}} \lambda_i(x) e_{l(i, sp^{(c)}(x))},$$

where $sp^{(c)}(x)$ is the shortest path from $o^{(c)}$ to $d^{(c)}$ at the x th iteration, and m is the number of commodities.

Our objective is to minimize $J(\lambda(x))/\alpha(\lambda(x))$. Let β be this minimum, i.e.,

$$\beta = \min_{\lambda(x)} \frac{J(\lambda(x))}{\alpha(\lambda(x))}.$$

The node weights are initialized as

$$\lambda_i(0) = \frac{\delta}{E_i}, \quad (17)$$

for all $i \in N$ as before. At each iteration x , for each commodity c , the shortest length path to the destination $d^{(c)}$ is at hand from the previous iteration, which is denoted by $sp^{(c)}(x-1)$. For each commodity c , an amount of $Q^{(c)} \Delta f(x)$ is augmented on its shortest length path, where $\Delta f(x)$ is calculated by

$$\Delta f(x) = \min_{c \in C} \min_{i \in \mathcal{N}_{sp^{(c)}(x-1)}^{(c)}} \frac{E_i}{e_{l(i, sp^{(c)}(x-1))}} \frac{1}{Q^{(c)}}, \quad (18)$$

and the total flow assigned until iteration x is updated by

$$f(x) = f(x-1) + \Delta f(x).$$

The node weights on the shortest path are updated as follows.

$$\lambda_i(x) = \lambda_i(x-1) \left\{ 1 + \frac{\epsilon}{m} \sum_{c \in C} \frac{Q^{(c)} \Delta f(x)}{E_i / e_{l(i, sp^{(c)}(x-1))}} \right\}, \quad (19)$$

for all $i \in \mathcal{N}_{sp^{(c)}(x-1)}$, where ϵ is a constant to be chosen later. Finally, the algorithm stops after t iterations, where $J(\lambda(t)) \geq 1$ for the first time.

There is a feasible flow of value $\frac{f(t)}{\log_{1+\epsilon} \frac{1+\epsilon}{\delta}}$, which can be shown as in the single commodity case. The analysis is the same as that in the single commodity case except for the running time. We have n nodes, and for each iteration, we need m shortest path calculations. Therefore, the running time is

$$n \left\lceil \frac{1}{\epsilon} - 1 + \frac{1}{\epsilon} \log_{1+\epsilon} n \right\rceil m T_{sp}, \quad (20)$$

which is $O(n(\frac{1}{\epsilon} \log_{1+\epsilon} n) m T_{sp})$.

4 Delay Constrained Maximum Lifetime Problem

In this section, we consider the problem with an additional constraint on delay. For simplicity, the single commodity case will be treated only. Extension to the multicommodity case will be straightforward.

The problem is defined as maximizing the system lifetime with the constraint that the delay between the origin node o and the destination node d be less than or equal to H , which can be written as

$$\begin{aligned}
 & \text{Maximize} && T \\
 & \text{s.t.} && \hat{q}_{ij} \geq 0, && \forall i \in N - \{d\}, \forall j \in S_i, \\
 & && \sum_{j \in S_i} e_{ij} \hat{q}_{ij} \leq E_i, && \forall i \in N - \{d\}, \\
 & && \sum_{j : i \in S_j} \hat{q}_{ji} + TQ_i = \sum_{k \in S_i} \hat{q}_{ik}, && \forall i \in N - \{d\}, \\
 & && \sum_{j \in \mathcal{N}_i} \tau_{l(j,i)} \leq H, && \forall i \in \{i \mid \hat{q}_{l(j,i)} > 0, \forall j \in \mathcal{N}_i\},
 \end{aligned} \tag{21}$$

where $\tau_{l(j,i)}$ is the delay from node j to its next hop on path i .

The problem above can be rewritten using the flow variables as before, and the primal problem is given by

$$\begin{aligned}
 & \text{Maximize} && \sum_{i \in P} f_i \\
 & \text{s.t.} && f_i \geq 0, && \forall i \in P, \\
 & && \sum_{j \in P_i} e_{l(i,j)} f_j \leq E_i, && \forall i \in N - \{d\}, \\
 & && \sum_{j \in \mathcal{N}_i} \tau_{l(j,i)} f_i \leq H f_i, && \forall i \in P.
 \end{aligned} \tag{22}$$

The dual problem of the above is given by

$$\begin{aligned}
 & \text{Minimize} && \sum_{i \in N} \lambda_i E_i \\
 & \text{s.t.} && \lambda_i \geq 0, && \forall i \in N, \\
 & && \eta_i \geq 0, && \forall i \in P, \\
 & && \sum_{j \in \mathcal{N}_i} \lambda_j e_{l(j,i)} + \eta_i \left(\sum_{j \in \mathcal{N}_i} \tau_{l(j,i)} - H \right) \geq 1, && \forall i \in P.
 \end{aligned}$$

We propose an algorithm which applies the same idea as in section 3.1. However, instead of incrementing the flow on the shortest path we use the delay constrained shortest path which is the shortest path of all $i \in P$ that satisfies $\sum_{j \in \mathcal{N}_i} \tau_{l(j,i)} - H \leq 0$. In other words, we are enforcing the delay constraints to be met in each iteration, and setting $\eta_i = 0$ for all i would result in a larger feasible region for λ_i 's, which in turn will yield a better optimization.

The description of the proposed algorithm is exactly the same as before except that the shortest path is replaced by the delay constrained shortest path.

The analysis of the performance and the running time is exactly the same as that in section 3.1 without the delay constraint. The one and most important

difference is that the problem of finding a delay constrained shortest path is NP-complete. The decision version of the problem is shown to be NP-complete in [1].

If we assume that the delay between any two nodes is fixed, say τ , then the delay constraint becomes the constraint on the number of hops which must be less than or equal to $\lfloor \frac{H}{\tau} \rfloor$. Since the shortest path obtained after x iterations in the Bellman-Ford algorithm provides the shortest path with at most x hops, the delay constrained shortest path problem in this case becomes solvable in polynomial time by limiting the number of iteration in the Bellman-Ford algorithm by that many iterations instead of $N - 1$ iterations.

5 Conclusion

In this paper, we have proposed fast approximate algorithms for the maximum lifetime routing problem in wireless ad-hoc networks in the single commodity case and in the multicommodity case with the analysis on the performance bound and the running time. In addition, an extended problem where there is an additional delay constraint has been studied. The delay constrained shortest path problem is NP-complete in general. However, the same algorithm can be used in the special case where the delay constraint is given as the maximum number of hops from the origin node to destination node by limiting the number of iterations in the Bellman-Ford algorithm.

References

1. Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows : Theory, Algorithms, and Applications*. Prentice Hall, 1993.
2. Dennis J. Baker and Anthony Ephremides. The architectural organization of a mobile radio network via a distributed algorithm. *IEEE Transactions on Communications*, COM-29(11):56–73, January 1981.
3. Jae-Hwan Chang and Leandros Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *Proceedings of IEEE INFOCOM2000*, Tel Aviv, Israel, March 2000.
4. Wai-Kai Chen. *Theory of Nets: Flows in Networks*. Wiley, 1990.
5. T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. McGraw-Hill and MIT Press, 1990.
6. Anthony Ephremides, Jeffrey E. Wieselthier, and Dennis J. Baker. A design concept for reliable mobile radio networks with frequency hopping signaling. *Proceedings of the IEEE*, 75(1):56–73, January 1987.
7. M. Ettus. System capacity, latency, and power consumption in multihop-routed SS-CDMA wireless networks. In *Proceedings of IEEE Radio and Wireless Conference (RAWCON) 98*, pages 55–58, Colorado Springs, CO, August 1998.
8. R.G. Gallager, P.A. Humblet, and P.M. Spira. A distributed algorithm for minimum weight spanning trees. Technical Report LIDS-P-906-A, Lab. Inform. Decision Syst., Massachusetts Inst. of Technol., Cambridge, MA, October 1979.

9. Naveen Garg and Jochen Koenemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proceedings 39th Annual Symposium on Foundations of computer science*, pages 300–309, Palo Alto, CA, November 1998.
10. D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. In Tomasz Imielinski and Hank Korth, editors, *Mobile Computing*, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.
11. Teresa H. Meng and Volkan Rodoplu. Distributed network protocols for wireless communication. In *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems, ISCAS'98*, volume 4, pages 600–603, Monterey, CA, June 1998.
12. A. Michail and A. Ephremides. A distributed routing algorithm for supporting connection-oriented service in wireless networks with time-varying connectivity. In *Proceedings Third IEEE Symposium on Computers and Communications, ISCC'98*, pages 587–591, Athens, Greece, June 1998.
13. S. Murthy and J.J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Journal of Special Topics in Mobile Networks and Applications (MONET)*, 1(2):183–197, 1996.
14. Vincent D. Park and M. Scott Corson. A highly distributed routing algorithm for mobile wireless networks. In *Proc. IEEE INFOCOM'97*, pages 1405–1413, Kobe, Japan, 1997.
15. C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers. In *ACM SIGCOMM*, pages 234–244, London, England, August 1994.
16. Volkan Rodoplu and Teresa H. Meng. Minimum energy mobile wireless networks. In *Proceedings of the 1998 IEEE International Conference on Communications, ICC'98*, volume 3, pages 1633–1639, Atlanta, GA, June 1998.
17. Timothy Shepard. Decentralized channel management in scalable multihop spread spectrum packet radio networks. Technical Report MIT/LCS/TR-670, Massachusetts Institute of Technology Laboratory for Computer Science, July 1995.
18. S. Singh, M. Woo, and C.S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proceedings of Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 181–190, Dallas, TX, October 1998.