Skyline Method In Wireless Ad-Hoc Networks Routing

[1] Fadoua YAKINE, [2] Manar ABOUREZQ and [3] Abdellah IDRISSI
Computer Science Laboratory (LRI)
Computer Science Department, Faculty of Sciences
Mohammed V University in Rabat
[1] yakine.fadoua@gmail.com [2] manar.abourezq@gmail.com [3] idrissi@fsr.ac.ma

Abstract—Unlike conventional routing protocols, in this paper, we address the problem of supporting unicast sessions with Quality of Service (QoS) requirements, using a new approach based on the Skyline operator. We have developed an algorithm that selects the routes while optimizing four characteristics related to routing, namely the number of hops, bandwidth, delay and cost. To evaluate the effectiveness and efficiency of our proposed algorithm, we compare its performance by simulation with an ILP algorithm based on a QoS constrained formulation.

Key-words: Wireless Ad Hoc Network; Routing; Unicast; QoS; Skyline; Block-Nested Loops Algorithm.

1 Introduction

A wireless Ad-Hoc Network is a decentralized network where nodes can communicate with each other without requiring any established infrastructure or centralized administration.

The main objective of an Ad Hoc network is to establish and maintain end-to-end connection and ensure reliable transport of data packets. The wireless Ad Hoc nodes are communication end-points as well as routers. To establish the communication between two non-direct neighbor end-nodes, sending packets by intermediate nodes is needed. Due to the dynamic nature of this kind of network, each node takes single-hop decisions based on the network topology [1].

Routing in wireless Ad Hoc networks is a very challenging issue. The design of good routing protocols should take into account the bandwidth limitation of the wireless spectrum, the power awareness and the multi-hop nature of such networks [2].

Many modern network applications such as transmission of multimedia data, real-time collaborative work, and interactive distributed applications require QoS provisions in order to work properly. Consequently, the QoS routing in Ad Hoc networks has received increasingly intensive attention [3].

Quality of service (QoS) refers to the network performance level provided by a service to the user. The main goal of QoS provisioning in communication network is to achieve a more deterministic network behavior so that information carried by the network at higher quality level and the use of network resources can be better utilized [4].

In this paper, we consider the problem of unicast routing with quality of Service (QoS) in wireless Ad Hoc networks, returning a least cost path between the source node and the destination node and satisfying a set of constraints that the path should respect. This problem is shown to be NP-complete when the path is subject to two or more constraints [5].

We chose a new approach to solve this problem, based on the Skyline operator [6]. The Skyline was introduced to solve problems that have complementary goals. The most prominent example [6] is that of selecting a hotel that would be both cheap and close to the beach. These two goals are complementary, because the closer a hotel is to the beach, the more expensive it is. The Skyline operator solves such dilemmas.

In the light of this, we propose to use this same approach in routing by computing the Skyline of the routes in a wireless network warranting QoS parameters like hop count, delay, bandwidth and cost. To evaluate the efficiency of our proposed algorithm, we compare its performance by simulation with another algorithm based on ILP

[31] (integer Linear Programming) approach on many network topologies.

Our paper is organized as follows. In the next section, we present some related work. In section 3, we present the Skyline operator. In section 4, we describe the model of the network we worked on along with the skyline Routing algorithm we used. Section 5 introduced the routing algorithm based ILP we proposed for making comparison between the two algorithms. Then in section 6, we present some results illustrating the implementation of our algorithm and finally we conclude in section 7.

2 Related works

A wide range of routing algorithms for Ad Hoc networks has been proposed in the literature [1, 7]. They had all in common: the maximization of the throughput while minimizing the packet loss, the control overhead and the energy usage. In [7], Boukerch et al. introduced a taxonomy of Ad Hoc protocols. They divided them in nine categories :(i) source-initiated (reactive or on-demand), (ii) table-driven (pro-active), (iii) hybrid, (iv) location-aware (geographical), (v) multipath, (vi) multicast, (vii) geographical multicast, (viii) hierarchical, and (ix) power-aware. This classification was based on their underlying architectural framework. A similar approach was taken in [1], the authors presented a survey of the routing algorithms (RA) for wireless networks. They classified the algorithms into various categories such as Geographical, Geo-casting, Hierarchical, Multi-path, Power-aware, and Hybrid routing algorithms. This survey offered an intensive study of those categories. It also compared, analyzed, and discussed the relationships among them and the routing issues that some of these algorithms try to solve.

Many studies have approached the problem of QoS routing [5, 8, 9]. Some algorithms have taken into account only one constraint. Plotkin discussed in [8] the competitive routing strategy with the consideration of the bandwidth only. The QoS routing with multiple constraints was also studied. The authors in [5] proposed an algorithm for constrained routing with bandwidth based on Dijikstra algorithm. Ma and Steenkiste proposed in [9] a modification of the Dijkstra algorithm for the computation of Widest-Shortest Paths or WSP algorithms to select the shortest path that is a feasible path according to the bandwidth constraint of flows [10]. In the objective to minimize the consumption of batteries power, Idrissi and al. presented in [11] a method based on an adaptation of the Dijkstra's algorithm to the MANET problem called MANED and proposed in [12] an approach based on an adaptation of the A star algorithm to minimize the consumption of energy.

In our previous papers [13, 14], we discussed the Multicast QoS routing problem with energy efficiency in wireless Ad Hoc networks. Multicast routing deals with finding a multicast tree, which is rooted from the source and addresses many destination nodes. In [13], we formulated the problem of multicast QoS routing as integer linear programming problem with a set of energy and QoS constraints. The proposed algorithm minimizes the total power of energy used by nodes while satisfying QoS constraints (Bandwidth and maximum delay) that are crucial to wireless Ad Hoc network performance. In order to prolong the network lifetime, we discussed the energy-efficient multicast problem in Ad Hoc networks with the respect of delay [14]. To do so, we formulated the problem as a Constraint Optimization Model for the Mobile Ad-Hoc Network problem along with delay constraint. For unicast communications, we have presented in [15] the Routing-IP model to optimize QoS routing in WANETs. This model reduces the energy consumption of nodes while meeting the QoS requirements in terms of end-to-end delay and bandwidth.

Many different approaches have been introduced for routing wireless networks. Our work was motivated by the Skyline operator [6] as it can help to make intelligent decisions over complex data when multiple requirements are considered. The Skyline operator was used in many recent works and different research areas, from the exploitation of databases to mobile Ad Hoc networks, and many more…

Babanejad et al. used the Skyline in [16] to search through database systems with growing data and find the results that best meet the user's preferences. They adapted the Skyline queries to dynamic databases with missing values. Kim and Yoon proposed in [17] a predictive algorithm based on the Skyline to use in recommendation systems. The proposed algorithm can be used to predict the user's rating and preferences when using or buying items online (articles, products, services…). Shang et al. presented in [18] a new query for the discovery of regions of interest, called the Path Nearby Cluster (PNC) query. The PNC is based on the principle of the Skyline and takes into account the route related requirements specified by the user. Zheng et al. proposed in [19] an approach to city planning based on the Skyline. In this work, the GPS trajectories of taxicabs are used to detect the regions that have traffic problems and compare the needs to the available resources such as roads, subway lines…

Yu and Bouguettaya presented in [20] a web services composition system based on the Skyline. The system selects the web services that are considered to be the most adapted to compose a composite service Skyline, taking into consideration different requirements such as the response time, the cost…

Huang et al. [21] presented techniques for Skyline query processing through mobile Ad Hoc networks, aiming to reduce communication cost and processing time on each single mobile device. Vlachou and Norvåg [22] studied bandwidth-constrained Skyline queries over mobile environments, which maximize the quality of the retrieved subset of the Skyline points based on a given constraint. Kriegel et al. adapted in [23] the Skyline to route computation in a road network considering multiple preferences. They proposed an efficient algorithm called ARSC, which is able to calculate all Pareto optimal paths in an efficient time.

We propose a new approach for routing in Ad Hoc networks based on the use of the Skyline operator as non-classical method to deliver messages. Authors in [29] have adapted this method for searching and selecting a service in the cloud services. Our approach proposed here consists in introducing the Skyline operator for routing in Wireless Ad-hoc Networks. As defined in [6] and described in [30], we will recall and present the principal of the Skyline operator in the next section.

3 The Skyline operator

The Skyline operator [6] was introduced to select the points that satisfy certain requirements that are contradictory. A classical example, as detailed in [6], is that of the choice of a hotel that is at the same time cheap and close to the beach. These two goals are complementary, since in general, the closer a hotel is to the beach, the more expensive it is. The Skyline operator allows solving such dilemmas by returning the hotels that are not dominated by any other hotel, that is to say hotels for which there is no hotel that is closer to the beach and cheaper.

The Skyline operator computes the Skyline, which is a set of points selected from a large set of data. These points are called tuples. A tuple p is formalized as $p = (p_1, p_2, \ldots, p_n)$ where pi is the value of the tuple p for the dimension i. A dimension is a criterion used to make the selection. In this hotel example, the dimensions are the cost and the distance to the beach.

The Skyline contains the tuples which are not dominated by any other tuple. Thus, each tuple in the Skyline is at least as good as any other tuple outside of the Skyline for all the dimensions, and better than any other point outside of the Skyline for at least one dimension.

Borzsonyi et al. proposed two major ways for implementing the Skyline. The first one is extending existing database systems. The second one is using algorithms [6].

Extending existing database systems is an intuitive way to compute the Skyline. It is done by using standard SQL instructions and adding a logical Skyline operator, SKYLINE OF, that is to be used to extend the SELECT clause. In this new clause, the dimensions that are used in the Skyline are specified and the operator MIN, MAX or DIFF is added to each dimension to specify whether the dimension is to be minimized, maximized or different. However, using SQL usually results in complex queries, especially when the number of dimensions exceeds three, which leads to a poor performance. In order to overcome it, the Skyline may be implemented using algorithms.

There are many algorithms that can be used, such as the Block-Nested Loops algorithm [6], the Divide-And-Conquer algorithm [22], the B-Tree algorithm [23], the R-Tree algorithm [24]… To implement the Skyline, we chose to use the BNL algorithm because it has a high performance, especially if the Skyline is small. Its complexity varies between $O(n)$ in the best case and $O(n^2)$ in the worst case, n being the number of tuples in the input list.

The BNL algorithm is an iterative algorithm that consists of comparing tuples among them to determine the ones that are not dominated by any other. It is done by keeping dominating tuples in the main memory and comparing each new tuple to them. In each iteration, a new tuple is read from the input list of tuples. There are three possible options:

1) If the new tuple is dominated by one of the tuples in the main memory, it is eliminated.

2) If the new tuple dominates a tuple in the main memory, the dominated tuple is eliminated, and the new tuple is added to the main memory to be compared to future tuples.

3) If the new tuple is incomparable, which means that it is neither dominated by nor dominating any tuple in the main memory, it is added to the latter.

At the end of all iterations, only tuples that are not dominated by any other tuple are kept in the main memory. These tuples form the Skyline.

We present our algorithm to compute the Skyline and the network we worked on in the next section.

4 Network Modeling

The wireless network is simply represented as weighted connected network G=(V,E), where V ={1,2,…N} is the set of nodes and E is the set of wireless links. In general, G is not a fully connected graph and the network supports multi-hopping where a message could travel over multiple hops to reach its destination. Each link e=(u,v) ∈ E has its properties: a delay Dl(e), bandwidth utilization Bd(e) and cost Co(e).

Delay function: $Dl(e) : E \rightarrow \mathbb{R}+$

Bandwidth utilization: $Bd(e) : E \rightarrow \mathbb{R}+$

Cost function: $Co(e) : E \rightarrow \mathbb{R}+$

The bandwidth utilization Bd(e) denotes the current traffic flowing through the link e. The delay D(e) represents the time needed to transmit information through that same link. Co(e) denotes the cost of using the link e.

We consider the unicast routing problem with hop count, bandwidth, delay, and cost constraints from one source node to one destination node and we use Skyline computation to find the best route from the source node s to a destination node d such that the bandwidth utilization, the delay, the cost and the hop count of this route are minimum.

Let $P_{(s,d)}$ denotes a unique path from the source node s to a destination node d.

A path $P_{(s,d)}$ from node s to destination node d is a sequence of nodes $(v_0,v_1,..,v_k)$ where the following conditions hold [27]:

$v_0=s$, $v_k=d$,

$v_i \neq v_j, (v_i,v_{i+1}) \in E$ (for $i \neq j, 0 \leq i \leq k-1$)

Our routing problem is defined by this metrics:

1) Hop count cost (H):

The basic idea of the hop count metric is simple and easy to implement. We assume that all links costs are equal to one unit, independent of the quality or other characteristics of the link.

The hop count H is defined as the length of the path $|P_{(s,d)}|$. It is the number of links (hops) a message performs before reaching the destination node [27].

$$H(P_{(s,d)}) = |< v_0 v_1 ... v_k >| \qquad (1)$$

2) Delay Path cost (D):

The total delay cost of the path $P_{(s,d)}$ is defined as the sum of the delay cost of all links in that path and can be given by [28]:

$$D(P_{(s,d)}) = \sum_{e \in P_{(s,d)}} Dl(e) \qquad (2)$$

3) Bandwidth Path cost (B):

The link bandwidth values are added to yield the bandwidth path cost (B).It can be expressed by [28]:

$$B(P_{(s,d)}) = \sum_{e \in P_{(s,d)}} Bd(e) \qquad (3)$$

4) Cost function (C):

The cost of each link may be associated with the cost of using the link. The cost of the path $P_{(s,d)}$ can be calculated by:

$$C(P_{(s,d)}) = \sum_{e \in P_{(s,d)}} Co(e) \qquad (4)$$

As for our routing algorithm, called Skyline Routing algorithm, it is performed in two sub-tasks. First, there's the computation of all paths $P_{(s,d)}$ going from the source node s to the destination d. Second, there's the choice of the best path among all paths based on Skyline computation.

Paths $P_{(s,d)}$ are calculated by a path finding-algorithm. The output is a file containing the list of all paths from s to d with their respective costs (H,D,B,C). The Skyline intervenes in the second step to provide the best route from the list previously obtained. We use the Skyline operator to choose the best paths that minimize the hop count, delay, bandwidth and cost.

We implemented the Skyline using the BNL algorithm, as showed in Fig. 1 hereafter and presented in [29, 30]. It takes as input a list of tuples for which the Skyline is to be computed. These tuples are the list of all paths $P_{(s,d)}$.It also uses as input the list of dimensions used by the Skyline. Each dimension has an indication if it is to be minimized, maximized or different. In our case, the dimensions are: Hop count, Delay path cost, Bandwidth path cost and path Cost (H,D,B,C). These dimensions are all to be minimized.

```
– L_I : input list of tuples for which the Skyline is to
  be computed
– L_D: input list of dimensions
– p, q: tuples
– L_O : output list of the tuples forming the Skyline
FunctionComputeSkyline
 Foreach p in L_I do
      If L_O = ∅ then
      L_O = {p}
      Else
      Foreach q in L_O– {p} do
      If p >> q then
      L_O = L_O+ {p} – {q}
      Elseif p << q then
      Goto (*)
      Elseif q is the last tuple in Ls
            L_O = L_O + {p}
      End IF
      End Foreach
     (*) End If
 End Foreach
 Return L_O
End Function
```

Fig. 1. The Skyline Routing algorithm

A new tuple p is read from the input list L_I ($P_{(s,d)}$) in each iteration. Then for each tuple p, a tuple q is read from the Skyline list. Then p and q are compared. If p dominated q, then q is eliminated from the Skyline list and p is added. If q dominates p, then p is disregarded and a new tuple is read from the input list L_I. If p and q are incomparable, then we move on to the next tuple in the Skyline list and compare it to p. If q is the last tuple, then p is incomparable to any tuple in the Skyline and is added to the Skyline list. This way, only dominating tuples are kept in the Skyline list at the end of all iterations.

The Skyline returns the paths that optimize the costs that are related to them, namely the number of hops, the delay bandwidth and cost. In the next section, we present the ILP based algorithm used to make the comparison between the two algorithms.

5 ILP Based Approach

We compared our Skyline approach with the ILP based approach inspired from our previous works in [13, 15]. It is based on Integer Linear Programming [31], an exact solution technique, to optimize QoS routing in Wireless ad hoc networks.

We adapt the model in [13] to the network modelling assumptions we presented in section 4.

Our proposed approach focuses on finding the optimal routes between the source node s and destination d given the QoS constraints namely: Hop count cost H, Delay path cost D, Bandwidth path cost B and cost function C.

We used a constrained formulation to solve the problem of unicast routing. The main goal is to construct a topology with the respect of QoS requirements. The objective of our proposed approach is to minimize hop-count, delay, bandwidth and cost constraints. The set of variables are defined in section 5.1. In section 5.2, we describe the four objectives we used in our formulation, while the topology constraints of the ILP Formulation for routing and the ILP algorithm are presented in sections 5.3 and 5.4 respectively.

5.1 Variables

In our formulation, we used the following variables:

- To represent a link between node i and node j, we define the Boolean decision variable X_{ij} as:

$$X_{ij} = \begin{Bmatrix} 1 \; if\ the\ link\ beetwen\ i\ and\ j\ exists \\ 0 \; otherwise \end{Bmatrix} \quad (5)$$

- To ensure the flow constraint , we define the Boolean decision variable $x_{i,j}^{s,d}$ as :

$$x_{i,j}^{s,d} = \begin{Bmatrix} 1 \; if\ there\ is\ a\ route\ from\ s\ to\ d\ throungh\ \ link\ (i,j) \\ 0 \; otherwise \end{Bmatrix} \quad (6)$$

5.2 Objective functions

We consider four different objectives functions.

The first objective of the ILP model we proposed is to minimize the hop count cost H. It is modelled as minimizing the summation of $\sum_{(i,j)} x_{i,j}^{s,d}$ of all links, as described in Equation (7):

$$\min H = \sum_{(i,j)} x_{i,j}^{s,d}, \forall s \in V, \forall d \in V \quad (7)$$

The second objective is to minimize the delay Path cost D. it is expressed as the minimization of the summation of $\sum_{i=1}^{N} y_i$ for all nodes, as described by (8):

$$\min D = \sum_{i=1}^{N} y_i \text{ where } y_i - Dl_{ij} X_{ij} \geq 0; \forall (i,j) \in V, i \neq j \quad (8)$$

Dl_{ij} is the delay associated with the link (i,j). y_i is the delay cost of each node i.

The third objective concerns the minimization of the Bandwidth Path cost B, that is expressed by the minimization of the summation of $\sum_{i=1}^{N} z_i$ for all nodes. It is given in equation (9):

$$\min B = \sum_{i=1}^{N} z_i \text{ where } z_i - Bd_{ij}X_{ij} \geq 0; \forall (i,j) \in V, i \neq j \quad (9)$$

Bd_{ij} is the bandwidth associated with the link (i,j). z_i is the bandwidth cost of each node i.

The fourth objective corresponds to the minimization of the cost of the path P(s,d). It can be written:

$$\min C = \sum_{i=1}^{N} w_i \text{ where } w_i - Co_{ij}X_{ij} \geq 0; \forall (i,j) \in V, i \neq j \quad (10)$$

where Co_{ij} is the cost associated with the link (i,j). w_i is the cost of each node i.

5.3 Topology Constraints of the ILP Formulation for Routing

The topology constraints of the ILP formulation for unicast routing should ensure valid topologies and valid routes between s and d. These constraints can be described by the following set of equations.

$$\left\{ \begin{array}{ll} \sum_{j=1}^{N} X_{ij} \geq 1; i = s, i \neq j & (11) \\ \sum_{j=1}^{N} X_{ij} = 1; \forall j \in D, i \neq j & (12) \\ \sum_{j=1}^{N} X_{ij} \leq (N-1)\sum_{j=1}^{N} X_{ji}; \forall (i,j) \in \{V - s\}, i \neq j & (13) \\ \sum_{j} x_{i,j}^{s,d} - \sum_{j} x_{j,i}^{s,d} = \begin{cases} 1 & \text{if } s = i \\ -1 & \text{if } d = i \forall i \in v \\ 0 & \text{otherwise} \end{cases} & (14) \\ x_{i,j}^{s,d} \leq X_{ij}; \forall i, j \in V & (15) \end{array} \right.$$

The constraint (11) expresses the condition that the source node s of the request should transmit at least once.

The node reachability constraints (12) express that any number of transmissions can be made out of a node i.

In order to avoid any loops in the final tree, constraints (13) are added .they stipulate that a node (except the source s) can transmit only if it receives a transmission from some other node to prevent any loops in the final solution.

Constraint (14) ensures that all links on (s,d) should meet the flows conservation.

Constraint (15) ensures that the route between each node-pair is valid. It states that traffic is circulating from node i to node j only when the link (i,j) exists.

5.4 The ILP algorithm

The following steps briefly describe the proposed algorithm:

- **Input**: Network V is a graph of N nodes, delay matrix Dl, bandwidth matrix Bd, cost matrix Co, and an unicast request set: R= {(*s*, *d*)}, s being the source node, d is the destination.
- **Output:** the optimal route from s to d with the 4 costs <H,D,B,C>.

1. **Run** the ILP model with previous equations (5)-(15) to compute the optimal route with the respect of the minimum hop count, delay, bandwidth and cost.
2. **Return** the optimal solution of the ILP.
3. **END**

In the next section, we present some results of the experimentations we conducted.

6 Experimentation and results

All the computational experiments are performed on a personal computer with 4 GB RAM and 2.4 GHz, Core 2 Duo Intel processor.

In order to test our skyline based approach, we implemented the path-finding algorithm in Matlab and the Skyline algorithm using ASP.net. To solve the routing algorithm based on the ILP formulation, we used the solver software CPLEX 12.5.0.0 [32] based on the branch-and-bound method in the same networks and report the results given from both of them.

6.1 Experiments 1: The Skyline Routing algorithm

The first experiments we have conducted were carried out on two-dimensional free space region as shown in Fig. 2.

To each edge of the network is assigned three weights Dl, Bd and Co.

We illustrate through Table 2 the results given by our algorithms to find the optimal path on the 10-node network.

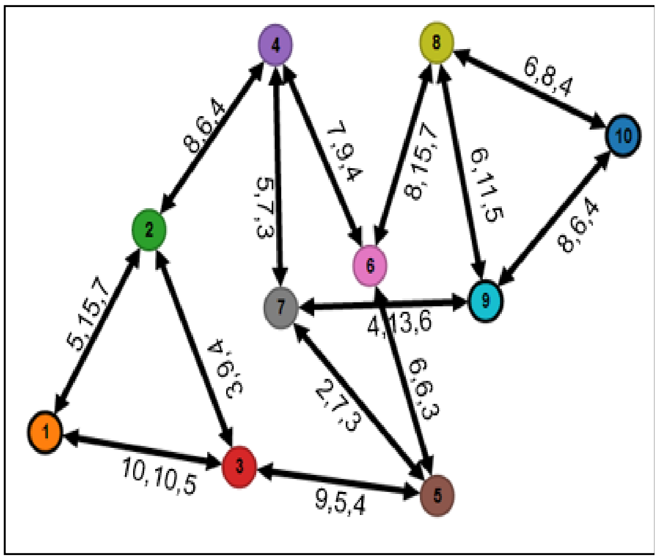

Fig. 2. The Input Network Topology With N=10 and Number of Edges = 14

We initiate 8 unicast requests in Table 2: The source s (2nd column) and destination d (3rd column). In the fourth and fifth column, we report the optimal routes found with their associated costs (5th column): hops, delay, bandwidth and cost. We first compute the path-finding algorithm that gives all the paths between s and d (see Table 1). Then we apply the Skyline operator on the paths found as explained earlier.

TABLE 1. THE OUTPUT OF THE PATH FINDING ALGORITHM FOR REQUEST N°=1 (S=2 AND D=4)

Path $P_{(s,d)}$	**$H(P_{(s,d)})$**	**$D(P_{(s,d)})$**	**$B(P_{(s,d)})$**	**$C(P_{(s,d)})$**
2-4	1	8	6	4
2-3-5-6-4	4	25	29	15
2-3-5-7-4	4	19	28	14
2-1-3-5-6-4	5	37	45	23
2-1-3-5-7-4	5	31	44	22
2-3-5-7-9-8-6-4	7	39	69	33
2-3-5-6-8-9-7-4	7	41	66	32
2-1-3-5-7-9-8-6-4	8	51	85	41
2-3-5-7-9-10-8-6-4	8	47	72	36
2-1-3-5-6-8-9-7-4	8	53	82	40
2-3-5-6-8-10-9-7-4	8	49	69	35
2-1-3-5-7-9-10-8-6-4	9	59	88	44
2-1-3-5-6-8-10-9-7-4	9	61	85	43

TABLE 2. SOME REQUESTS AND THEIR ROUTES

Req n°	**S**	**d**	**Number of paths**	**Optimal Routes**	**<$H(P_{(s,d)})$,$D(P_{(s,d)})$, $B(P_{(s,d)})$, $C(P_{(s,d)})$>**
1	2	4	13	1	<1,8,6,4>
2	2	8	18	2	<4,23,37,18> <3,23,30,15>
3	6	5	10	1	<1,6,6,3>
4	10	7	14	1	<2,12,19,10>
5	1	10	32	4	<5,33,41,22> <5,30,47,24> <5,39,44,23> <5,34,53,26>
6	7	6	8	2	<2,8,13,6> <2,12,16,7>
7	3	9	18	1	<3,15,25,13>
8	5	2	8	1	<2,12,14,8>

The second simulations were performed in a larger networks with N=50, 100, 200 and 400 nodes. We varied the number of dimensions from 1 to 4 and the input size of routes from 100 to 10 000, depending on the number of nodes. Ten instances are generated and solved for each of these values and average results are reported. The results are showed in Tables 3, 4, 5 and 6.

TABLE 3. AVERAGE SOLVING TIMES (MS) WITH DIFFERENT INPUT SIZE VALUES WITH N=50

Number of dimensions	***Input size***	***Average Execution time (ms)***	***Number of optimal routes***
1	100	10	1
	500	9.8	1
	1000	10.7	1
	2 000	6	1
	6 000	10	1
	8 000	9	1
	10 000	12	1
2	100	11	1
	500	12.2	1
	1000	12.8	1
	2 000	8	8
	6 000	15	8
	8 000	14	8
	10 000	18	8
3	100	11.6	9
	500	13	10
	1000	14.3	11
	2 000	9	8
	6 000	13	8
	8 000	17	8
	10 000	19	8
4	100	8	8
	500	8	8
	1000	8	8
	2 000	11	8
	6 000	17	8
	8 000	19	8
	10 000	22	8

TABLE 4. AVERAGE SOLVING TIMES (MS) WITH DIFFERENT INPUT SIZE VALUES WITH) N=100

Number of dimensions	*Input size*	*Average Execution time (ms)*	*Number of optimal routes*
1	100	5	1
	500	5	1
	1000	8	1
	2 000	6	1
	4 000	8	1
2	100	6	1
	500	6	1
	1000	6	1
	2 000	7	1
	4 000	9	1
3	100	6	1
	500	8	1
	1000	7	1
	2 000	8	1
	4 000	9	1
4	100	7	1
	500	8	1
	1000	8	1
	2 000	9	1
	4 000	8	1

TABLE 5. AVERAGE SOLVING TIMES (MS) WITH DIFFERENT INPUT SIZE VALUES AND WITH N=200

Number of dimensions	*Input size*	*Average Execution time (ms)*	*Number of optimal routes*
1	100	5	1
	500	5	1
	1000	6	1
2	100	6	3
	500	6	3
	1000	6	3
3	100	6	5
	500	7	5
	1000	7	5
4	100	7	5
	500	8	5
	1000	9	5

TABLE 6. SAME THAN BEFORE FOR N=400

Number of dimensions	*Input size*	*Average Execution time (ms)*	*Number of optimal routes*
1	100	5	1
	500	5	1
2	100	6	3
	500	6	3
3	100	6	5
	500	7	5
4	100	7	5
	500	8	5

6.2 Experiments 2: The Skyline Routing algorithm vs. ILP Routing algorithm

In Experiments 2, we compared the Skyline Routing and the ILP Routing algorithms for large number of nodes up to 400 and report the results given from both of them.

The simulation results given are reported in Tables 7 and 8.

Our algorithm based on the Skyline method gives better routes or at least the same routes as with the ILP based approach on the network topology given on Fig. 2 (see Table 7).

TABLE 7. COMPARISON BETWEEN THE ROUTES OF SKYLINE COMPUTATION AND ILP BASED APPROACH

Req n°	**s**	**d**	**Number of paths**	**Skyline**		**Optimal approach**	
				Optimal Route	**<H,D,B,C>**	**Optimal Route**	**<H,D,B,C>**
1	2	4	13	1	<1,8,6,4>	1	<1,8,6,4>
2	2	8	18	2	<4,23,37,18> <3,23,30,15>	1	<4,23,37,18>
3	6	5	10	1	<1,6,6,3>	1	<1,6,6,3>
4	10	7	14	1	<2,12,19,10>	1	<2,12,19,10>
5	1	10	32	4	<5,33,41,22> <5,30,47,24> <5,39,44,23> <5,34,53,26>	1	<5,30,47,22>
6	7	6	8	2	<2,8,13,6> <2,12,16,7>	1	<2,8,13,6>
7	3	9	18	1	<3,15,25,13>	1	<3,15,25,13>
8	5	2	8	1	<2,12,14,8>	1	<2,12,14,8>

In this experiment, we aimed to compare the execution time of our algorithm to the execution time of the ILP algorithm for the same networks of 50, 100, 200 and 400 nodes. In Table 8, we gave the execution times relative to the ILP based approach and the skyline approach for different values of number of dimensions. Each of the experiments in this section was repeated many times. The results are the average value of ten instances generated for each dimension

TABLE 8. COMPARISON BETWEEN THE ROUTES OF SKYLINE COMPUTATION AND ILP BASED APPROACH

Network size	*Number of Dimensions*	**ILP based approach**		**Skyline approach**	
		Average Execution Time (ms)	*Number of Optimal Routes*	*Average Execution Time (ms)*	*Number of Optimal Routes*
N=50	1	560	1	9.4	1
	2	560	1	11.48	4
	3	570	1	14.2	8,85
	4	596	1	13.28	8
N=100	1	2840	1	6.4	1
	2	2850	1	6.8	1
	3	2750	1	7.6	1
	4	2806	1	8	5
N=200	1	15723	1	5.33	1
	2	15830	1	6	3
	3	15496	1	6.66	5
	4	14736	1	8	5
N=400	1	55853	1	5	1
	2	48406	1	6	3
	3	46896	1	6.5	5
	4	50323	1	7.5	5

7 Conclusion

Routing of packets is a challenging task to accomplish efficiently in Ad Hoc networks due to their dynamic nature and limited resources. We propose, in this paper, a new approach to unicast routing based on the use of the Skyline operator that warrants QoS parameters. As seen in the experimentation section, our approach gives some interesting first results compared with the ILP based approach and can be used as an alternative to classical routing approaches.

References

[1] E. Alotaibi and B. Mukherjee, "A survey on routing algorithms for wireless Ad-Hoc and mesh networks", Computer Networks 56.2, (2012): 940-965.

[2] G. Parissidis, M. Karaliopoulos, R.Baumann, T.Spyropoulos and B.Plattner, "Routing metrics for wireless mesh networks", In Guide to Wireless Mesh Networks. Springer London, (2009):199-230.

[3] H. Badis, I. Gawedzki, and K. Al Agha, "QoS routing in Ad Hoc networks using QOLSR with no need of explicit reservation", In Vehicular Technology Conference, VTC2004-Fall. IEEE 60th. IEEE, (2004): 2654-2658.

[4] TB. Reddy, I. Karthigeyan, BS .Manoj and CSR .Murthy, "Quality of service provisioning in Ad Hoc wireless networks: a survey of issues and solutions", Ad Hoc Networks 4.1 (2006): 83-124.

[5] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications", Selected Areas in Communications, IEEE Journal on 14.7 (1996): 1228-1234.

[6] S. Börzsönyi, D. Kossmann, and K. Stocker, "The Skyline operator", International Conference on Data Engineering (ICDE), (2001): 421-430.

[7] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Bölöni, D. Turgut, "Routing protocols in Ad Hoc networks: A survey", Computer networks 55 (2011):3032–3080.

[8] S. Plotkin, "Competitive routing of virtual circuits in ATM networks", Selected Areas in Communications, IEEE Journal on, vol.13, no.6, (1995):1128-1136.

[9] Q. Ma, P. Steenkiste, "Quality-of-Service Routing for Traffic with Performance Guarantees", Building QoS into Distributed Systems. Springer US, (1997): 115-126.

[10] M. Curado and E. Monteiro, "A survey of QoS routing algorithms", in Proceedings of the International Conference on Information Technology (ICIT 2004), Istanbul, Turkey, (2004).

[11] A. Idrissi, "How to minimize the energy consumption in mobile ad-hoc networks", In International Journal of Artificial Intelligence & Applications, (2012).

[12] A. Idrissi, C.M. Li and J.F. Myoupo, "An Algorithm for a Constraint Optimization Problem in Mobile Ad-hoc Networks",18th IEEE International Conference on Tools with Artificial Intelligence, ICTAI'06, Arlington, VA, USA, (2006): 555-562.

[13] A. Idrissi and F. Yakine, "Multicast routing with quality of service constraints in the Ad Hoc wireless networks", In Journal of Computer Science, 10, (2014):1839-1849.

[14] F. Yakine and A. Idrissi, "Delay-constrained efficient multicast routing in wireless Ad Hoc networks", In proc. of Next Generation Networks and Services (NGNS), 2014 Fifth International Conference on. IEEE, (2014): 289-294.

[15] F. Yakine and A. Idrissi, "Energy-Aware Topology Control And Qos Routing In Ad-Hoc Networks", Procedia Computer Science, (2015):309-316.

[16] G. Babanejad, H. Ibrahim, N. I. Udzir and F. Sidi, "Finding Skyline Points over Dynamic Incomplete Database", In Proceedings of Malaysian National Conference on Databases (MaNCoD), (2014).

[17] S. Kim and Y. Yoon, "Recommendation system for sharing economy based on multidimensional trust model", Multimedia Tools and Applications, (2014): 1-14.

[18] S. Shang, K. Zheng, C. S. Jensen, B. Yang, P. Kalnis and J. R. Wen, "Discovery of Path Nearby Clusters in Spatial Networks", Knowledge and Data Engineering, IEEE Transactions on 27.6 (2015): 1505-1518.

[19] Y. Zheng, Y. Liu, J. Yuan and X. Xie, "Urban computing with taxicabs", In Proceedings of the 13th international conference on Ubiquitous computing , ACM, September (2011):89-98.

[20] Q. Yu and A. Bouguettaya, "Efficient service Skyline computation for composite service selection", Knowledge and Data Engineering, IEEE Transactions on 25.4 (2013): 776-789.

[21] Z. Huang, C.S. Jensen, H. Lu and B.C. Ooi, "Skyline queries against mobile lightweight devices in MANETs", in Proceedings of the 22nd International Conference on Data Engineering (ICDE), (2006):66-66.

[22] A. Vlachou and K. Nørvåg, "Bandwidth-constrained distributed Skyline computation." In Proceedings of the International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE), (2009):17–24.

[23] H.-P., Kriegel, M. Renz and M. Schubert, "Route Skyline queries: A multi-preference path planning approach", In IEEE 26th International Conference on Data Engineering (ICDE), (2010): 261-272.

[24] H. Kung, F. Luccio, and F. Preparata, "On finding the maxima of a set of vectors", Journal of the ACM (JACM) 22.4 (1975): 469-476.

[25] D. Comer, "The Ubiquitous B-Tree", ACM Computing Surveys (CSUR) 11.2 (1979): 121-137.

[26] N. Beckmann, H. P. Kriegel, R. Schneider and B. Seeger, "The R*-tree: an efficient and robust access method for points and rectangles", ACM, Vol. 19, No. 2, (1990):322-331.

[27] R. Beraldi and R. Baldoni, "Unicast routing techniques for mobile Ad Hoc networks", In Handbook of Ad Hoc Networks, CRC Press, New York, (2003).

[28] S. Chen and K. Nahrstedt, "On Finding Multi-Constrained Paths", Proc. IEEE Int'l Record on Comm. (ICC '98), (1998): 874-879.

[29] A. Idrissi and M. Abourezq, "Skyline in Cloud Computing", Journal of Theoretical and Applied Information Technology 60.3 (2014).

[30] M. Abourezq and A. Idrissi, "Introduction of an outranking method in the Cloud computing research and Selection System based on the Skyline", Proceedings of the International Conference on Research Challenges in Information Science (RCIS), (2014).

[31] A. Schrijver, "Theory of linear and integer programming", John Wiley & Sons, (1998).

[32] IBM ILOG CPLEX Optimizer, http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/index.html