A Delay Constrained Routing Algorithm for LEO Satellite Networks

Zongyang Luo, Zhili Sun and Haitham Cruickshank

***Abstract*—In the future personal communication systems, LEO satellite networks will be used to provide real time multimedia services. Every satellite in this system has routing capability with the application of on-board processor. As one of the most important technologies, routing is indeed a hard problem in the LEO satellite networks due to movement of satellites and changes of the network topology. Many algorithms have been proposed to solve this problem but most of them do not consider how to provide QoS based routing while tackling the basic routing problem. In this paper, we use a weighted graph model to analyze the QoS routing problem in LEO satellite networks and propose a Delay Constrained Longest lifeTime (DCLT) routing algorithm for LEO satellite networks. Our algorithm tries to find the paths that fulfill the delay constraint of users requests using an approximation methodology. At the same time, it tries to find a path that has longest lifetime, which is more efficient since the re-routing attempts are reduced. We use both theoretical analysis and simulation to study the performance of the algorithm we proposed.**

***Index Terms*—QoS routing, LEO satellite networks**

I. Introduction

AS an important part of future communications infrastructure, satellite networks will play a very important role. Compared with geostationary satellites (GEO), low earth orbit (LEO) satellites have their unique characteristics, such as low propagation delay and low requirement for the ground terminals' power supply. With the on-board processors equipped on satellites, some advanced functions, i.e., routing, become possible. Networking LEO constellations reduces the dependence on the ground stations and make the LEO satellite networks becomes a relatively autonomy system [1, 2].

For multi-hop satellite networks, routing is a very challenging problem both in academic and engineering fields because of the high dynamic nature of LEO satellite networks. The topology of LEO satellite networks is changing all the time, which makes networks' state update information flooding in the network. Also the long propagation of inter satellite link (ISL) and high error code rate will make the update information inaccurate. All of these make the current routing algorithms and protocols in terrestrial networks cannot be applied in LEO satellite networks directly. On the other hand, more and more real time applications make the quality of service (QoS) become more and more important. QoS routing (QoSR) provides a possible approach to solve this problem.

In this paper, we propose a QoS source routing algorithm, which meets the delay requirement of every routing request by using the path with longest lifetime. For every routing request, the source node adopts an approximation algorithm to choose all the paths that meet its delay requirement. From these paths, the one which has the longest lifetime is chosen. The rest of this paper is organized as follows. In section II, we introduce the background and related work. In section III, we present the weighted graph model and describe our algorithm in detail. Simulation results and results analysis are covered in Section IV and we will conclude this paper in section V.

II. Procedure for Paper Submission

A lot of work has been carried out to solve the routing problem in LEO satellite networks.

For connection-oriented applications, Markus [3] has proposed a routing algorithm called Discrete Time Dynamic Virtual Topology Routing designed for ATM based LEO satellite networks. The algorithm divides the satellite's period into limited discrete time slots. It computes the routing table for all the nodes in advance and makes optimization work when the time slots change. Seong [4] has proposed an algorithm combining the routing problem and topology design. It models the LEO satellite networks as a Finite State Automation (FSA) using satellite constellation information. The topological design deals with the selection of ISLs and the routing handles the traffic distribution over the selected links to maximize the number of carried calls.

As for connectionless applications, Ekici [5] has proposed a distributed routing algorithm, Distributed Datagram Routing (DDR). The aim of DDR is to minimize the packet delay in the network. In this algorithm, one geography coordinate is used to represent the satellites' locations in the network. Narvaez [6] has proposed an IP routing framework, Internet Routing on Satellite Networks, (IRSN), for satellite networks. In this framework, only the entry and exit point operate the traditional IP routing functions, the specific routing algorithm for satellite networks is used in the whole LEO satellite networks.

The researchers mentioned above have done good investigation work for LEO satellite networks. However, they did not consider the following aspects. The first one is the delay problem in LEO satellite networks. LEO satellite networks are supposed to support real time multimedia applications in the future. Low packet transmission delay is

The authors are with the Centre for Communication Systems Research, University of Surrey, Guildford, Surrey, GU2 7XH. U.K. (email: Z.Luo@surrey.ac.uk; Z.Sun@surrey.ac.uk; H.Cruickshank@surrey.ac.uk).

crucial for both service providers and end users. On the other hand, due to the high mobility of satellite nodes, paths chosen may fail very soon after it begins to transmit packets. Choosing a path with longer effective time can reduce the number of re-routing operations.

In order to alleviate the above problem, we apply QoS routing algorithms in LEO satellite networks. In traditional Internet, researchers have done deeply work to try to find algorithms for QoS routing with better performance. The main aim is to find a path that satisfies the QoS requirements for every request and balance the resource utilization of the networks. Some of the main difficulties in QoS routing problem are as follows. First, the computation complexity of QoSR is NP-Complete (NPC). Second, the performance of the algorithm highly depends on the network state information. As for LEO satellite networks, it is different from the traditional Internet. It is in a high mobility environment, which means the network state and topology change very fast and this also makes the routing computation overhead much heavier as well. However, the topology of LEO satellite networks can be predicted and computed in advance. With this characteristic, some routing computation work can be performed off line in advance. As for QoSR in LEO satellite networks, Junka [7] has proposed a distributed QoS routing algorithm (DQR). DQR chooses the path with longest lifetime which meets the delay requirement. The distributed implementation has its nature disadvantage for LEO satellite networks. Firstly, it is difficult to guarantee the robustness of distributed routing algorithm because some routes may fail in some locations of the topology [8]. Secondly, distributed routing algorithms usually have long routing convergence process, which means the path's lifetime become shorter and this makes re-routing attempts more frequent. Here we propose a delay constrained source routing algorithm for LEO satellite networks to overcome the defect of distributed routing algorithm.

III. Delay Constraints Longest Lifetime Routing Algorithm

A. LEO satellite networks architecture and weighted graph model

We consider the LEO Satellite networks as a polar-orbiting constellation, composing of N planes and each plane has M satellites. Every satellite has four Inter Satellite Links (ISLs), which include two inter-plane links, the links between the satellites in the different plane, and two intra-plane links, the links between the satellites in the same plane.

Two important parameters are considered in our routing algorithm, one is called *lifetime* and the other is link delay. We use lifetime to represent the dynamic characteristics of the links between satellites in the constellation, which is similar to the one proposed by Junka [7]. We define lifetime as the existing period of link between every two satellites. Different satellites links have different lifetimes. The intra-plane links are connected all the time, thus the lifetime is considered as infinite. However, the links on different planes might be broken as they are not maintained regularly. In the low latitude area, the lifetime of inter-plane link is long, but as the latitude increases, it decreases dramatically. The inter-plane link will be temporarily switched off in the polar area which means that the lifetime of the links become zero. The changes of inter-plane link's lifetime will impact the routing performance in the LEO satellite networks. Another important parameter is link delay. It is defined as the propagation time of the packets via laser link. The link delays of inter-plane and intra-plane have different characteristics. In inter-plane links, the delays change quickly all the time. Long delay occurs when the satellite is in low latitude area, it becomes shorter as the satellites move into higher latitude area. When the satellites move into the polar area, the inter-plane links are switched off for a moment and the link delay is considered as infinite. As for the intra-plane satellite links, they are connected together all the time and the relative positions are fixed, thus the delay of intra- plane links can be treated as constant.

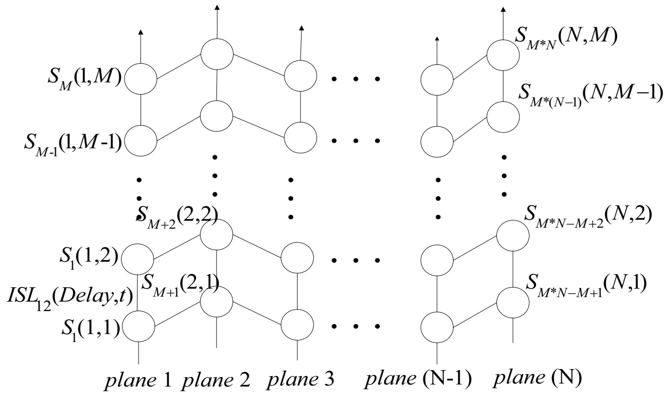

Fig. 1. Weighted graph of LEO satellite networks

We use a weighted graph $G(V,E)$ to represent the snapshot of the LEO satellite networks at some time points (Refer to Fig. 1). Nodes (V) of the graph represent the satellites in the LEO satellite networks and Edges (E) represent the links between two satellites. Different columns represent different planes in the network and the nodes in the same column represent the satellites in the same plane. We mark the satellite planes from plane 1 to plane N, and satellites from satellite 1 to satellite $M\times N$. Every link in the graph has a state measured by the QoS metrics of concern. As mentioned above, we consider the delay and lifetime in this paper. We use $d(E_{i,j})$ to represent the delay of link between node i and node j, as defined by equation (1). We use another metric, $t(E_{i,j})$, to represent the lifetime of a link. It equals to the inverse of Lifetime of link between node i and node j, as equation (2). We assume t is a kind of cost of the link between two satellites. Both of these two metrics are considered as additive metrics.

$$d(E_{i,j}) = Delay(E_{i,j}) \tag{1}$$

$$t\ (E_{i,j}) = \frac{1}{Lifetime(E_{i,j})} \tag{2}$$

Given a source satellite and a destination satellite, there could be a path, $P(S_s\text{->}S_d)$, connect between them. The delay of the path is defined as the sum of all the links' delays on the path by (3). The t of the path is defined as the sum of the t of all the links on the path by (4).

$$d(P(S_s->S_d))=\sum_{E(S_i,S_j)\in P(S_s->S_d)} d(E(S_i,S_j)) \qquad (3)$$

$$t\ (P(S_s->S_d))=\sum_{E(S_i,S_j)\in P(S_s->S_d)} t\ (E(S_i,S_j) \qquad (4)$$

If a delay constraint D is given, there could be more than one path fulfill the requirement with different t. Some of the paths might break quickly because the short path lifetime, while some of them might last longer due to long path lifetime. Path with longer lifetime means that it is more stable and needs fewer re-routing operations, which is very important for routing convergence in the LEO satellite networks.

Given D, finding a path with minimum t from the source satellite, S_s, to destination satellite, S_d, is a DCLC problem in graph theory, which is a NP hard problem. A lot of solutions have been proposed to solve this problem, but they did not consider the special topology for applications. In other words, they do not care much about the relationship between the network's topology and computational complexity. In usual network, it is very difficult to determine the topology, especially for mobile network or large size network. Under this condition, most of QoS routing algorithms can only use some assumptive parameters to perform the routing calculation process. But LEO satellite networks have a predicable topology, which can help us to optimize the QoS routing algorithm, reduce the computational complexity and re-routing operation. In next sub-section, we will explain our algorithm in details.

B. Delay Constraint Longest LifeTime (DCLT) source routing algorithm for LEO satellite networks

First we define the exact problem we are going to solve. Given a LEO satellite weighted graph $G(V,E)$, which includes M orbits, N satellites per plane. Every satellite has two intra-plane links and two inter-plane links. Thus there are $K=M\times N\times 2$ ISLs. Here, we assume the LEO satellite networks is all connected and the links' failure or network seams of some LEO satellite constellations can be represented by cost. We mark all the ISLs with ISL_k, where k=1,2,…,K. The delay of ISL_k is d_k and the cost of ISL_k is t_k. The delay constraint of the path is D. Our objective is to find a path whose delay is shorter than delay constraint, D, and it has the minimal cost t. We can use function (5) and function (6) to represent this objective. If a path meets both of these two functions, it is the solution.

$$f_1=D-\sum_{k=1}^{K} d_k x_k \geq 0 \qquad (5)$$

$$f_2=Min(\sum_{k=1}^{K} t_k\times x_k) \qquad (6)$$

$$x_k=\begin{cases}1, ISL_k\in\{P(S_s->S_d)\}\\ 0, ISL_k\notin\{P(S_s->S_d)\}\end{cases}, k=1,2,...,K \qquad (7)$$

We use x_k to represent the link states between every two nodes. If there is a link between two nodes, it equals to 1, while if there is no link or the link is switched off between two nodes, it equals to 0.

As mentioned above, this is a DCLC problem in graph theory, which has one constrained parameter and one optimized parameter. It is possible to add more parameters to this problem, such as delay jitter, etc. Then it will be a MCOP problem, which has one constraint and more than one metrics to be optimized. For the problem we define, we propose an approximation algorithm to solve it, which is based on the algorithm proposed by Chen [9] and Korkmaz [10]. Instead of using a fixed approximation method they proposed, we use a dynamic approximation method according to the specific characteristics of the LEO satellite networks. We define the exact approximation parameter range to reduce the computational complexity.

First we create a new delay function from R_0^+ to G=[1, g] as follows:

$$d'(ISL_k)=\left\lceil\frac{d(ISL_k)\times g}{D}\right\rceil \qquad (8)$$

where g is a given positive integer, D is the delay constraint, R_0^+ is the set of non-negative real number. We define g as:

$$g=coefficient\times L \qquad (9)$$

where *coefficient* is a positive integer, to be used to adjust the precision of our algorithm and L is the number of links in the path chosen.

Therefore, function (5) can be converted to:

$$f_3=g-\sum_{k=1}^{K} d'_k\, x_k \geq 0 \qquad (10)$$

With the mapping function (8), the original problem's computational complexity has been reduced from exponential to polynomial and the solution to the simpler problem must be a solution to the original one. The proof can be found in [9].

This algorithm reduces the computational complexity, and at the same time it also reduces the accuracy of the solution, which means there may be a solution to the original problem but we can not find it using the algorithm we proposed. Thus we must analyze the condition where we can balance between the computational complexity and algorithm's accuracy. On the condition that if D, g and L are given, the solution to equation (10) is also the solution for equation (5), but this is only a sufficient condition, not a necessary condition. The restriction is given as [9]:

$$d(P)\leq(1-\frac{L-1}{g})\times D \qquad (11)$$

The approximation parameter, g, can decide the performance of this routing algorithm. In order to make solution to the new problem close to the original one, g must be set large enough. However, as g increases, it affects the computational complexity significantly. Therefore, we must estimate the number of ISLs in path and choose the suitable value of L to minimize the computational complexity of source node.

In usual network toplogy, it is hard to estimate L, because the topology is irregular. Thus we usually choose the longest distance, L_{max}, between any two nodes. In other words, it is a fixed approximated parameter under any condition, which can induce high computing overhead under some unnecessary conditions and waste the computing resource.

Here we try to use the LEO satellite networks' specific features, which are the regular topology and periodicity. We can make use of these features and choose L_{max} felicitously. Assuming that we have all the satellites' positions information in advance. Whenever a routing request is initiated from the source satellite, the destination satellite can be in one of the following three conditions:

1) The destination satellite is in the same plane with the source satellite;

2) The destination satellite is in the ring composed of neighbouring satellites with only inter-plane ISLs;

3) The destinations are out of range mentioned in the above condition 1 and 2.

Based on the analysis above, we can conclude that the maximum number of the ISLs in the loop free path in LEO satellite networks is:

$$L_{\max} = \begin{cases} \max\{|j_d - j_s|, (N - |j_d - j_s|)\}, i_s = i_d \\ \max\{|i_d - i_s|, (M - |i_d - i_s|)\}, j_s = j_d \\ \max\{|i_d - i_s|, (M - i_d - i_s|\} + N \\ \quad - \min\{|j_d - j_s|, (N - |j_d - j_s|)\}, i_s \neq i_d, j_s \neq j_d \end{cases} \tag{12}$$

So if we know the source satellite and destination satellite in advance, we can choose suitable L_{max} according to the relationship of these two nodes and reduce the computational complexity effectively.

We can use the modified Dijkstra Shortest Path First (SPF) [12] algorithm to solve this problem in polynomial time. It can be described as follows.

Step 1: Use the satellites' topology information to judge the relative position of the source satellite S_s and destination satellite S_d;

Step 2: Choose the suitable L_{max} and set g;

Step 3: For each $k \in [1, g]$, using the modified Dijkstra Shortest Path First find the path fulfill the requirement.

Here we use source routing strategy. It requires the source satellite maintain the state information of the network, which can be done by link state protocol.

IV. Simulation Results And Analysis

In this section, we present the simulation and analysis of our algorithm. We use two important metrics to evaluate our algorithm's performance, one is success ratio, and the other is re-routing probability. First we give out the parameters of our simulation. We adopt the Iridium-like [11] LEO satellite networks topology as our simulation topology, which has 66 satellite nodes and 121 ISLs. The link delay and cost are set according to the Iridium constellation parameters. The parameter, t, is the inverse of the lifetime of the link. As for the inter-plane ISLs in the polar area, the lifetime of the links is zero because they are switched off, and t is infinite. In the equatorial area, the lifetime is the longest for the inter-plane ISLs, so t is the smallest. The t for intra-plane ISLs are constants. We assign the corresponding t to every link in the graph G. For each routing request, the value of delay constraint, D, is set to 50ms, 100ms, and 150ms respectively.

First, we discuss our algorithm's success ratio (SR). It is defined as

$$\text{SR} = \frac{\text{Number of Successfully Routed}}{\text{Total Number of Routing Request}}$$

Recall the parameters in the equations above, the approximation parameter, g, is decided by the source satellite and destination satellite's location and precision we need. When a routing request is initiated from one source satellite, the source satellite judges the relationship between the source satellite and destination satellite. If both of these two satellites are in the same plane, L_{max} is assigned a value of 10, if they are in the ring composed only by the neighboring satellites, it will be assigned to 5. In other cases, it is assigned to 15. We adjust the coefficient to make the result of our algorithm close to the exhaustive algorithm. The bigger the coefficient, the closer the result of our algorithm will be to the exhaustive algorithm.

First we choose three values for the delay constraints, which are 50ms, 100ms, 150ms. The routing requests are randomly generated. Every point in the Fig. 2 is taken by 100 routing requests. Then we discuss how our algorithm can reduce the computational complexity in the LEO satellite networks. We also fulfill one exhaustive algorithm, which can find all the optimal paths if they exist.

When the delay constraint is 50ms, the routing success ratio of the exhaustive algorithm is about 0.216. When coefficient is small, e.g., 1, which means g is 5, 10 or 15 according to the relationship between source node and destination node, the average routing success ratio is around 0.159. Therefore it has large probability that the routing

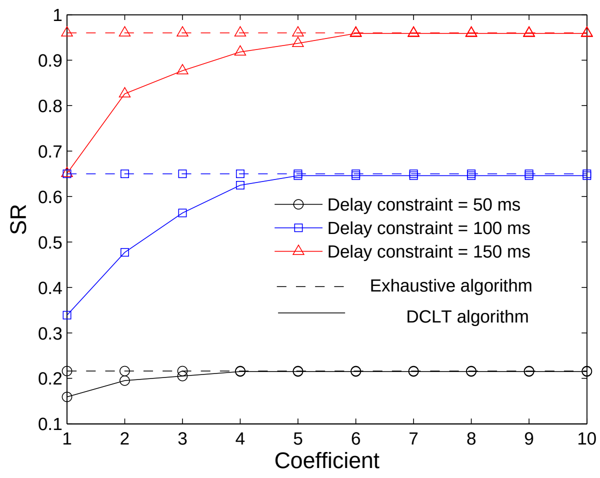

Fig. 2. SR of exhaustive algorithm and DCLT algorithm when delay constraints are 50ms, 100ms and 150ms.

request is refused by the source routing satellite. As the coefficient increases, the routing success ratio increases accordingly. When it increases to about 4, the success ratio of our algorithm is almost the same as the exhaustive algorithm.

When the delay constraints are set to 100 ms and 150 ms respectively, we observe that the routing success ratio increases as well. This means more paths can be found when constraint becomes more relaxed.

Next we compare the re-routing performance between DQR and our algorithm with metric, Re-routing Probability (RP). It is defined as

$$RP = \frac{\text{Average Number of Re-routing}}{\text{Total Number of Routing Request}}$$

Average number of re-routing is the statistical average value of re-routing attempts during all routing requests. 100 routing requests are proceed with different path distances, which is statistical numbers of ISLs in one path. If there was fewer re-routing attempts in one path, the path will have longer lifetime. From Fig. 3, we can see that our algorithm have lower RP than DQR, which means more frequent re-routing attempts of DQR compared with DCLT in the same session duration.

Finally we briefly discuss the computational complexity of our algorithm. Our algorithm is based on the Dijkstra Shortest Path First algorithm (SPF), the adopted data structure will affect the computation complexity. In our implementation, we use adjacent list. With this data structure, SPF has a computational complexity of $O(n^2)$. For every $k \in [1, g]$, we must check all the nodes in the graph, so the searching space size is $(g \times n)$. For every node in this searching set, the operation of finding the path with least delay and minimal t need $(g \times n)$ times, so in total, the computational complexity is $O(g^2 n^2)$. By setting suitable approximation parameter range, we reduce the value of g efficiently, consequently it can reduce the computational overhead of the source satellite in the LEO satellite networks.

V. Conclusion

In this paper, we propose a Delay Constrained Longest lifeTime (DCLT) source routing algorithm for LEO satellite networks. The aim of this algorithm is to find a path with optimization parameter, t, while fulfilling the delay requirement requested by the users. We use an approximation method to improve the performance of exhaustive algorithm. The approximated parameters' ranges are decided by the specific satellite constellation. With minimal t, the path can have longest effective time so that the re-routing attempts are reduced effectively. Through simulation, we observe that our algorithm has acceptable routing success ratio with low re-routing attempts.

In our work, we consider the ingress/egress points of the routing path are fixed satellites, which is not realistic because LEO satellites are moving around the earth all the time. Combined with dynamic ingress/egress points in our algorithm is one part of our work in the future. Another important point is that we use traditional graph theory to analyze the QoS routing algorithm in LEO satellite networks, which is based on a static graph model. How to use a new dynamic graph to model the network's dynamic characteristics and design multi-constraint routing algorithm for it in such kind of high dynamic network environment is a challenging problem and is also for our future research work.

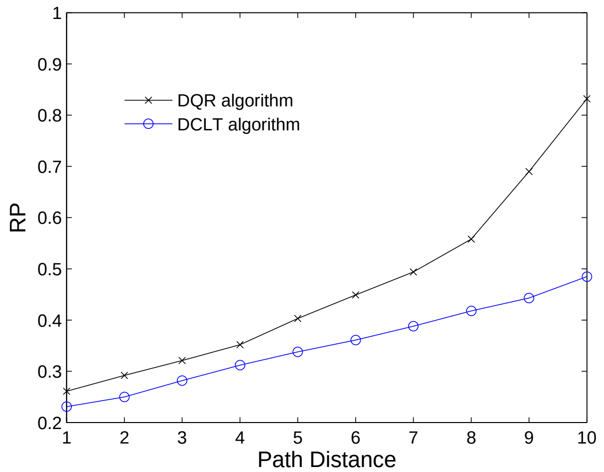

Fig. 3. RP comparison between DQR and DCLT

Acknowledgment

The authors sincerely thank the support of the IST SatNEx project.

References

[1] L. Wood, et al., "IP Routing Issues in Satellite Constellation Networks", Intrenational Journal of Satellite Communication, Vol. 19, No. 1, pp.69-92, John Wiley & Sons, January/February 2001.

[2] Z. Sun, M. P.Howarth, H. Cruickshank, S. Iyengar and L.Claverotte, "Networking Issues in IP multicast over Satellite", International Journal of Satellite Communications, Vol. 21, Number 4~5, pp. 489-507, July~October 2003.

[3] M.Werner, "Dynamic Routing Concept for ATM-based Satellite Personal Communication Networks", IEEE Journal on Selected Areas in Communications, Vol. 15, No. 8, pp. 1636-1648, Octorber 1997.

[4] H.S.Chang et al., "Topological Design and Routing for Low Earth Orbit Satellite Networks", Proc. Globecom '95, pp.529-535, 1995.

[5] E. Ekici, I. F. Akyildiz, and M. D. Bender, "A Distributed Routing Algorithm for Datagram Traffic in LEO Satellite Networks", IEEE/ACM Transactions on Networking, Vol. 9, No. 2, pp.137-147, April, 2001.

[6] P. Narvaes, A. Clerget, and W. Dabbous, "Internet Routing over LEO Satellite Constellations", WOSBIS'98, Proceedings of ACM/IEEE Workshop on Satellite-Based Information Services, Dallas, Texas, October 1998.

[7] A. Jukan, H. N. Nguyen and G. Franzl, "QoS-based routing methods for multi-hop LEO satellite networks", Networks, 2000, (ICON 2000), Proceedings, IEEE International Conference on, pp. 399-405, Sept. 2000.

[8] T.R. Henderson, and R.H. Katz, " On Distributed, Geograph-Based Packet Routing for LEO Satellite Networks", Proceedings of IEEE Globecom 2000, San Francisco, CA, Nov. 2000.

[9] S. Chen, K. Nahrstedt, "On Finding Multi-Constrained Paths", Communications, 1998, ICC 98. Conference Record, 1998 IEEE International Conference on, Vol. 2, pp.874-879, June 1998.

[10] T. Korkmaz. M. Krunz, and S. Tragoudas, "An efficient algorithm for finding a path subject to two additive constraints", Proceeding of the ACM SIGMETRICS 2000 Conference, Vol. 1, pp. 318-327, June 2000.

[11] R. J. Leopold., A. Miller, "The IRIDIUM communications system", Potentials, IEEE, Vol. 12, Issue. 2, pp. 6-9, April 1993.

[12] E.W. Dijkstra. A Note on Two Problems in Conexion with Graphs. Numerische Mathematik, 1:269-271, 1959.