Performance Study of Adaptive Routing Algorithms for LEO Satellite Constellations under Self-Similar and Poisson Traffic

Ioannis Gragopoulos, Evangelos Papapetrou, Fotini-Niovi Pavlidou

Aristotle University of Thessaloniki, School of Engineering
Dept. of Electrical & Computer Engineering, Telecommunications Division,
54006 Thessaloniki, Greece,
Tel:+ 30 31 996285, email: grag@egnatia.ee.auth.gr, epapet@egnatia.ee.auth.gr,
niovi@vergina.eng.auth.gr

ABSTRACT

A comparative study of routing techniques is carried out for LEO constellations interconnecting high speed terrestrial networks assuming *Poison* and *Self-Similar* input traffic. Shortest path routing as well as optimal routing (flow deviation) methods are applied for balanced and unbalanced traffic load and for uniform and non uniform distribution of the earth stations. The performance of flow deviation method is proved to be very successful even for the LEO complicated networks for both *Poisson or Self-Similar* input. A modification of the classic flow deviation algorithm regarding the number of paths we work with is proposed. Indeed, it is proved that a *k-paths flow deviation* method is always easy to obtain and gives robust results for any traffic pattern at a very affordable algorithmic complexity.

I. INTRODUCTION

The main function of a routing algorithm is the appropriate selection of a path (route) for any origin-destination pair on a network. Routing techniques for telecommunications networks have been studied extensively from early 60's in terms of the main performance measures, throughput (quantity of service) and delay of messages (quality of service). For low or moderate traffic flow, throughput is equal to the offered traffic so delay is the only essential measure, and so far this is the situation for current LEO satellite constellations [1].

There are a number of ways to classify routing algorithms. Centralized (all route choices are made at a central node) versus distributed (routing decisions made locally), static (fixed routes regardless traffic conditions) versus adaptive/dynamic (routes responding to traffic conditions), shortest paths versus optimal routing. In the following we shall focus our research on the category of optimal routing, investigating their influence on the network performance [2].

Most of the practical algorithms are based on the concept of a shortest path between two nodes. Here, each communication link is assigned a positive number called its length and between any pair of nodes we try to define the route of the shortest length. If the links are of unit length the shortest path is simply a minimum hop path. Shortest path routing has two drawbacks. First, it uses only one path per origin destination pair and second its capability to adapt to changing traffic conditions is limited by its susceptibility to oscillations. An improvement to this approach is the selection of *k*-shortest paths for any pair of nodes to be used according to the network load.

Optimal routing, [2,3] is based on the optimization of an average delay-like measure of performance, splitting any origin-destination pair traffic to many links and shifting traffic gradually between alternate paths, thus resulting in a more balanced distribution of the load on the network. Its application gives always a better performance than that of the shortest path algorithms for simple topology networks. But if the network to be studied is of a considerable complexity a thorough evaluation of the time/memory demands of the algorithms must be done. So far for LEO constellations a few contribution have been appended in the literature concentrating mainly on adaptive and non-adaptive shortest path algorithms [5,11].

In the following sections a comparative study will be presented for Shortest Path (SP) and optimal routing algorithms for complicated

LEO constellations. In section II the description and modeling of the system will be presented, in section III numerical results of extended algorithmic applications will be given for *Poisson* and *Self-Similar* traffic models and in section IV conclusions of our research and points of further investigation will be indicated.

II. SYSTEM MODELING

A. System description

A system of N=63 satellites in 7 circular orbits at a height of 1400 km is examined. The system is very close to LEO systems proposed in the literature and especially to the Celestri configuration [1]. An extended study on the topology of the system based on the azimuth and elevation angles of the constellation resulted in the selection of six InterSatellite Links (ISLs) at every node (satellite) on the network [4]. The network topology is changing continuously through the movement of satellites but its connectivity remains constant, the only changing feature being the length of the links.

The constellation is used for the interconnection of terrestrial high speed networks, so the traffic is bursty and of considerable intensity and the time constraints for some services are quite strict. In Table 1 are given some values of the selected system.

We considered two models for the terrestrial traffic distribution:

a) Earth stations are uniformly distributed on the earth surface, which is a very general assumption. The destinations are chosen randomly through a uniform distribution. This results in a very balanced load on the network.

b) Earth stations are gathered at some places on the earth surface [5] leading so to non uniform distribution of origin-destination pairs.

The load offered to the network by every earth station is taken out of a normal distribution with mean value μ and variance σ. If the variance is taken equal to zero all the earth stations offer equal throughput resulting in a balanced network load.

B. Routing Algorithms Selection

Flow Deviation Algorithm

From the class of optimal routing the Flow Deviation (FD) algorithm was selected for application [8]. The FD algorithm splits the load to different paths according to the path length given by a flow dependent metric. It continuously adapts this load splitting following the changes in path length trying to minimize the cost function given below. A cost function adaptive to the transmission delay is chosen using the well known formula [2]:

$$D = \frac{1}{\gamma}\left(\sum_{(i,j)} \frac{f_{ij}}{C_{ij} - f_{ij}} + p_{ij} f_{ij} \right) \quad (1)$$

Where $p_{i,j}$ is the transmission delay on link (i,j), $f_{i,j}$ is the flow on link (i,j), C_{ij} the capacity of link (i,j) and γ the total traffic offered to the network. The length of the link is taken equal to the derivative:

$$\frac{\partial D}{\partial f_{ij}} = D' \qquad (2)$$

The FD routing algorithm that we have adopted for the selected network has been tested for terrestrial networks giving very good performance [10]. Assuming a network of N nodes let W be the set of origin-destination pairs w. For each pair w a number of distinct paths N_p connect the origin to the destination node. The flow of each path is denoted by x_p and the resulting vector $x=\{x_p\}$ corresponds to the network routing pattern. The objective of the routing algorithm is to find a routing pattern x that minimizes the cost function D of Eq. (1). The algorithm iterates deviating flow from non-optimal to optimal paths until the routing pattern is optimized. This deviated amount is adjusted through a parameter called step-size and denoted by $a_s \in [0,1]$. For every iteration of the algorithm the value of α_s is adapted according to the following equation:

$$a_s = \min\left[1, \frac{\sum_{(i,j)} \left(\bar{f}_{ij} - f_{ij}\right) D'_{ij}}{\sum_{(i,j)} \left(\bar{f}_{ij} - f_{ij}\right)^2 D''_{ij}} \right] (3)$$

The steps of the Flow Deviation (FD) algorithm are the following:

Step 1. Find an appropriate number of paths for each origin-destination pair w of the network [10].

Step 2. With the mean value of the input traffic of each node find the flow of each link f_{ij} by using the last known routing pattern $x=\{x_p\}$ of path flows. Compute each link length from Eq. (2).

Step 3. Compute the initial value of the cost function Eq. (1) D_{step3}.

Step 4. For every pair w, find the length of each path and compute the shortest path. The path length is equal to the sum of the corresponding link lengths.

Step 5. Let $\bar{x} = \{\bar{x}_p\}$ be the routing pattern of path flows that would result if all input traffic of each pair w is routed along the corresponding shortest path. By using $\bar{x}$ compute the virtual flows $\bar{f}_{ij}$.

Step 6. For each network link compute the first and second derivatives D'_{ij}, D''_{ij} of the cost function. Compute the step-size α_s from Eq. (3).

Step 7. ***(a)*** Set origin node i=1.

(b) Let K be the number of distinct paths of the pair witch origin with node i. Set k=1.

(c) Deviate flow according to the following equation:

$$x_{i_k} = x_{i_k} + \alpha_s\left(\bar{x}_{i_k} - x_{i_k}\right)$$

where x_{i_k} is the flow of the k^{th} path of the pair with origin the node i.

(d) Set $k=k+1$. If $k>K$ go to ***Step 7(e)***, else go to ***Step 7(c)***.

(e) Set $i:=i+1$. If $i>N$ go to ***Step 8***, else go to ***Step 7(b)***.

Step 8. With the new routing pattern x that results from ***Step 7*** compute the new link flows f_{ij} of the network and compute the new value of the cost function D_{step8} from Eq. (1).

Step 9. If D_{step3}-$D_{step8} \le \varepsilon$ **END PROCEDURE**, else go to ***Step 2***. Where ε the desirable convergence error.

The application of equations (1), (2) and (3) assumes that the arrival pattern is a *Poisson* one, which is a very poor assumption for high speed networks. Recent studies and measurements of traffic on several networks have proved that self-similar modeling is more reliable and reasonable for traffic characterization [9].

Dijkstra Algorithm

Three standard SP algorithms are referred in the literature [6,7]: *Belman-Ford, Dijkstra* and *Floyd-Warshall*. The first two find the SP from all nodes to a given destination, while the third one finds the SP from all nodes to any other node. The computational requirements of these algorithms is out of the scope of this study since we examine their routing capabilities and not their implementation complexity. All these algorithms result to the same routing pattern for the same given conditions. We have selected the *Dijkstra* algorithm as the most popular of the three algorithms, (it works only with positive link lengths but this is the case in communications networks).

The *Dijkstra* algorithm has been applied in our study in two versions: a) considering the length of a link equal to the distance (propagation delay $p_{i,j}$) between the two nodes the link connects and b) taking the link length equal to a metric that depends on the link flow (adaptive *Dijkstra*). The most appropriate selection for the second case is to consider the link length equal to the quantity of Eq. (2).

The three routing algorithms, simple *Dijkstra*, adaptive *Dijkstra* and the FD algorithm have been applied to the selected topology for a real-time simulation method through the following procedure: The input traffic for each node is measured for the period of T_r slots and the cumulative result is fed in to the routing algorithm together with the previous routing pattern x. The algorithm decides for the new routing pattern and the procedure is repeated every T_r time slots. The frequency of the recall of routing algorithms is very critical for the network behavior as it will be shown in Section III.

C. *Input Traffic*

As it was mentioned above *Self-Similar* traffic modeling is more suitable for high speed networks offering a variety of multimedia services. Among the different proposals for modeling *Self-Similar* traffic we have selected the one proposed in [9].

This self-similar process is a mathematical model for the superposition of an infinity of on-off sources with *Pareto* distribution. Each source, at scarce random moments of time, begins to generate bursts of a random number of packets. The burst length distribution is *Pareto*. At the end of the burst generation, the source becomes silent for a random time which is, as a rule, greater than the length of the packet generation interval. The number of individual sources is so large that it can be considered as infinite but the total intensity λ of the sources is finite and of a given value. Under these conditions the suggested mathematical model for the aggregate network traffic is as follows:

We assume that the sources produce packets with a constant rate R. If ω_s is the s-th instant of the beginning of packet generation with the constant rate R and τ_s is the length of the interval where this generation takes place, the random variables τ_s are mutually independent, independent of ω_s and are identically distributed with *Pareto* type distribution:

$$P\{\tau_s \le t\} = 1 - \left(\frac{1}{t}\right)^{\alpha} \qquad (4)$$

Where and $0<\alpha$ and $1<t<\infty$.

The random variables:

ξ_t={number of time moments ω_s such that $\omega_s=t$} form a *Poisson* process with intensity λ, and they are independent of τ_s. Let Y_t be the total rate of packet generation in the aggregate traffic at time t. The process Y_t is assumed to be stationary. At time t, the packet generation jumps with value $R\cdot\xi_t$ up and falls down with value $R\cdot\kappa_t$, where κ_t is the number of active periods τ_s, terminated at time t. For this process the mean rate of packet generation is given by:

$$\mathrm{E}\{\mathrm{Y}_t\} = R * \lambda * \alpha_t \qquad (5)$$

and variance is given by:

$$\mathrm{E}\left\{(\mathrm{Y}_t - \mathrm{E}\{\mathrm{Y}_t\})^2\right\} = R^2 * \lambda * \alpha_t \qquad (6)$$

where α_t is the mean of the *Pareto* distribution.

III. NUMERICAL RESULTS

Any earth station is connected to a satellite node, which is continuously changing (handoff). Also, the six inter-satellite links are changing. A detailed description of the modeling and operation of the proposed system is given in reference [11]. These changes happen in a completely predictable way, so we can suggest that we examine different *static* topologies of the system. We also assume that the handoff is perfect, that is the flows are transferred from one satellite to another with no bandwidth problems.

The comparison of the algorithms have been accomplished through two different simulation techniques: a) non real-time and b) real-time implementations. As it was mentioned previously, the routing algorithms are executed periodically and the inputs at every running are a) the previous routing scheme and b) the mean value of the offered load. For non-real time simulation we assume some mean values for the offered load (produced by a normal distribution (μ, σ)) and then we execute only once the algorithms and we compute the mean delay of the network through Eq. (1). In other words the non real-time simulation corresponds to an instance of the real-time simulation. Via this approach we can compare the convergence capability of the algorithms and the obtained mean delay, but we can not study different kinds of input traffic. So in order to apply the *Self-Similar* traffic model it is necessary to apply a real-time simulation technique. Under real-time simulation, we mean that we examine the system at the packet level, packets generated according to different traffic models (*Poisson* and Self-*Similar*). Every single packet uses the routing pattern produced by the above routing algorithms and the routing pattern is updated every time period equal to T_r. In the following subsections we give results of both ways of simulation (non real-time and real-time). The comparison of the three proposed algorithms leads to some very interesting results.

Non real-time simulation

Since the FD algorithm converge after some iterations it is quite helpful to study first the parameters influencing its performance. In Fig. 1, we study the iterations needed for the convergence of the algorithm with parameter the mean load of the system, for a symmetric load (variance zero) and for symmetric distribution of the earth stations. Convergence is found to be late (more iterations) for heavy load but it is achieved in any case. The percentage of the load in the figures, for example load 20%, means that the input traffic of each station is the 20% of the links capacity.

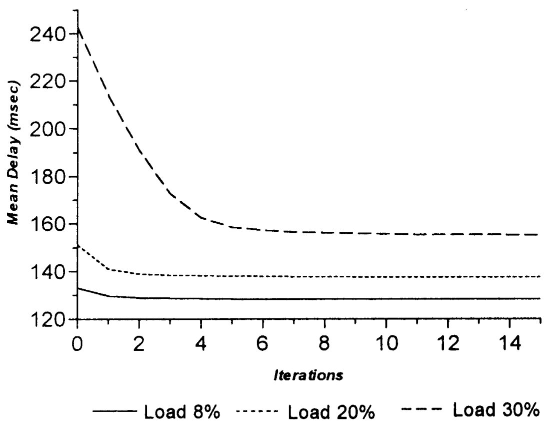

Figure 1. FD convergence for symmetric network load.

Since the possible paths for any source destination pair are numerous we chose some paths to work with. We denote N_p the number of alternative paths for each origin-destination pair. In Fig. 2 we give the influence of the number of selected paths on the performance of the algorithm, again studying the mean delay versus convergence. The choice of a limited number of paths, can be considered a very serious modification of the FD algorithm, and it is proved to be very efficient on complicated network topologies, reducing the time complexity of the application. Thus we can propose a new category of routing algorithms the *k*-shortest path flow deviation. Of course the more paths we use the less delay we obtain, and so a complete flow deviation is always superior than any *k*-shortest path, but investigating the final profit in delays we see that a considerable improvement is noted between a 3 and a 6 paths selection. For more that 6 paths, the improvement in delay does not compensate the complexity of the algorithm. We make clear that the selection of the alternative paths can be done through various criteria. We have applied here a minimum hop algorithm for the selection of the six alternative paths.

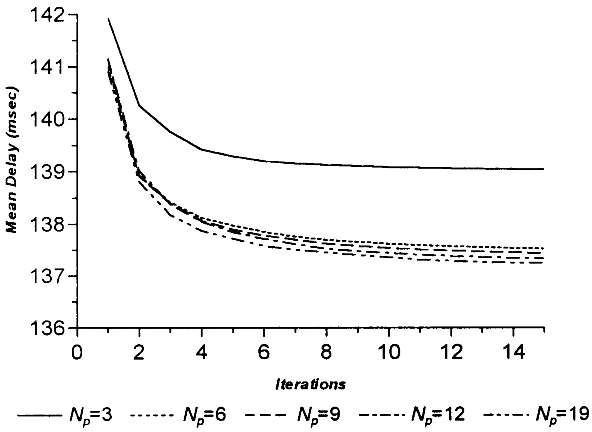

Figure 2. FD convergence according to N_p for symmetric network load.

A very interesting input is given in Fig. 3. The mean delay is studied for two cases: a) any origin-destination pair uses 6 alternative paths, the minimum hop paths selected as in Fig. 2 and b) any origin-destination pair uses 6 alternative paths, each one stemming from a different ISL leading to 6 disjoint paths. The performance is better in the first case leading to the conclusion that the intersatellite links may be not used at the time.

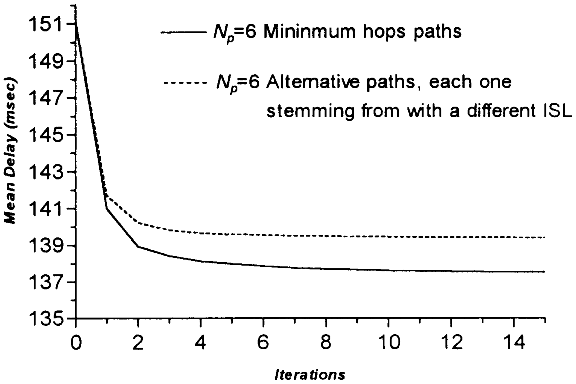

Figure 3. FD convergence according to N_p for different selection criteria.

In Fig. 4 we study the performance of FD for the unbalanced traffic case. We observe that despite an increase in the variation of load we obtain the same satisfactory performance of FD (assumed convergence to the same mean delay value even if the starting traffic values are very different).

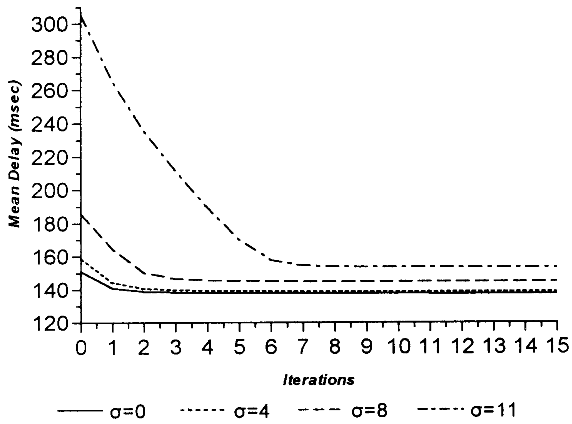

Figure 4. FD convergence according to different amount of variation of unbalanced network load with mean value of 18%.

In Figs. 5 and 6 we compare the three algorithms referred in section II. Delay versus load for the balanced and unbalanced network cases. As we expected the performance of *Dijkstra* is failing at heavy load and what is important it fails for unbalanced situations which is the permanent status for satellite constellations. The modified *Dijkstra* (adaptive *Dijkstra*) which is adaptive to traffic changes performs better, but the *k*-shortest path FD is always still superior.

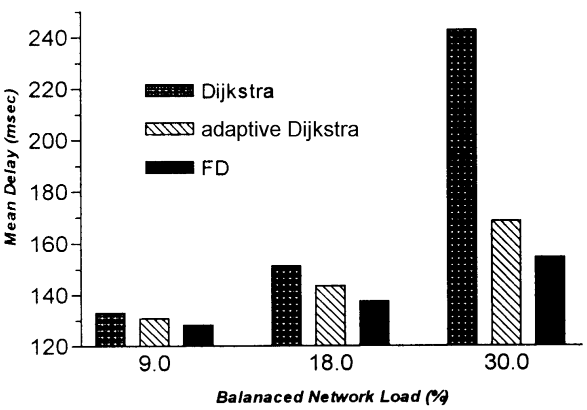

Figure 5. FD, *Dijkstra* and adaptive *Dijkstra* according to different amounts of balanced network load.

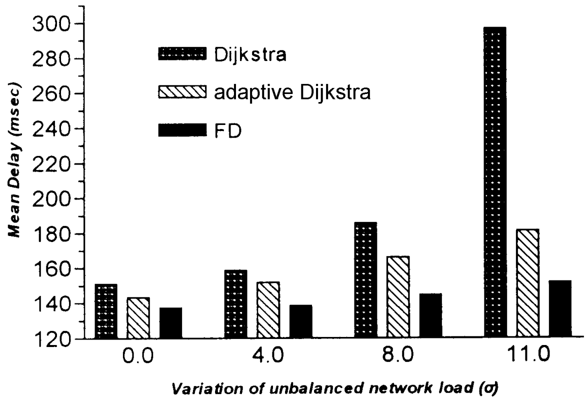

Figure 6. FD, *Dijkstra* and adaptive *Dijkstra* for the case of unbalanced network load according to variation σ, with mean network load 18%.

In Fig. 7 the case of non-uniform distribution of earth stations is examined. The load of the stations is assumed to be balanced. In the same figure the case of uniform distribution of earth stations is also displayed. The second bar of similar pairs corresponds to the non-uniform case of the corresponding algorithm. For light input load the three algorithms present no significant difference from the uniform case. But for heavy load conditions the simple *Dijkstra* algorithm fails to route the traffic. The other two algorithms present similar behavior in the case of uniform and non-uniform earth stations distribution, but they increase slightly the mean delay of the network.

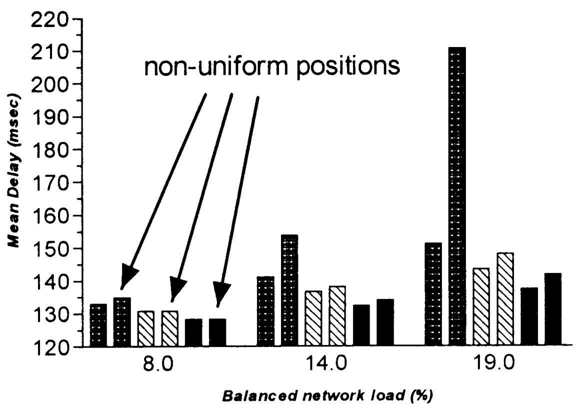

Figure 7. FD, *Dijkstra* and adaptive *Dijkstra* for balanced network load and non-uniform distribution of earth stations.

Real-time simulation

We will compare the three algorithms through a real-time simulation assuming *Poisson* and *Self-Similar* input traffic. In both cases we assume balanced network load and uniform distribution of ground stations. The network uses N_p=6 paths for every pair *w*. The values presented in the figures are measured in timeslots. A timeslot is a period of time that corresponds to an ATM cell (53 bytes).

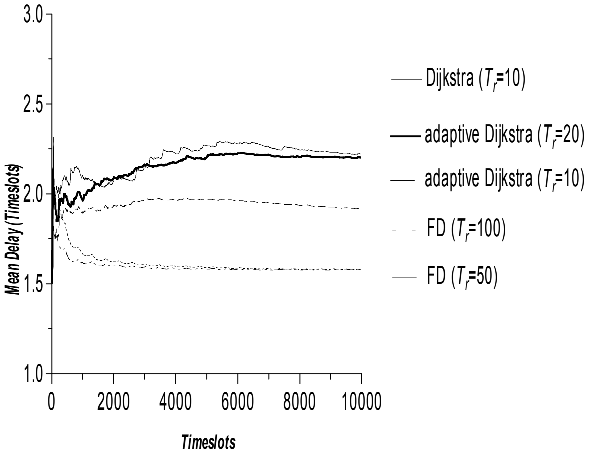

Figure 8. Real-time simulation for the *Poisson* case for the FD, *Dijkstra* and adaptive *Dijkstra* with link flow algorithms with mean input load 0.4 packets/timeslot.

In Fig. 8 the case of *Poisson* input traffic with mean value of 0.4 packets/timeslot considering the three algorithms is examined. All the algorithms manage to route the traffic and present stable performance during a time interval of 10000 timeslots. The FD algorithm results in the lowest mean delay with a maximum routing interval T_r=100. In other words with the FD algorithm we can achieve a better result even if the network executes the FD routing algorithm 10 times less frequently than the other two. If we decrease the routing interval T_r=50 for the FD algorithm the performance has no practical improvement since the *Poisson* environment is a very affordable choice for the FD algorithm. On the other hand the adaptive *Dijkstra* presents great degradation with an increase of the routing interval (T_r=20) and its performance is similar to the simple *Dijkstra* case.

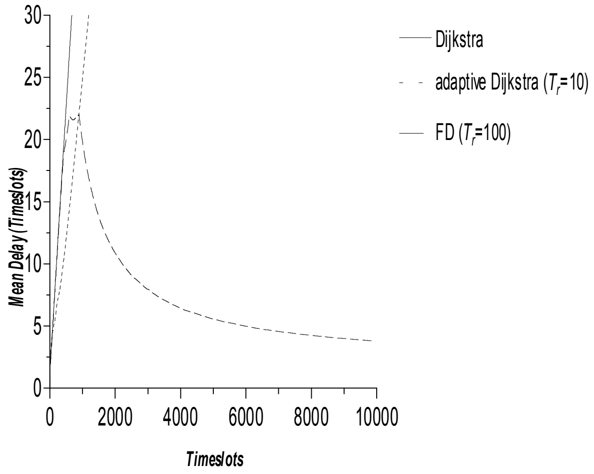

Figure 9. Real-time simulation for the Poisson case for the FD, *Dijkstra* and adaptive *Dijkstra* algorithms with mean input load 0.7 packets/timeslot.

In Fig. 9 the mean value of *Poisson* input traffic is increased to 0.7 packets/slot. The results are very impressive. The *Dijkstra* and adaptive *Dijkstra* fail to route the load but the FD conserve the mean delay to very low values splitting the input traffic to different paths. For the duration of 10000 timeslots we assume that all the sources offer traffic with density 0.7 packets/Timeslot. This assumption is not very realistic (too much input traffic for such a long time interval), but indicates the capabilities of the FD algorithm to exploit the resources of the network.

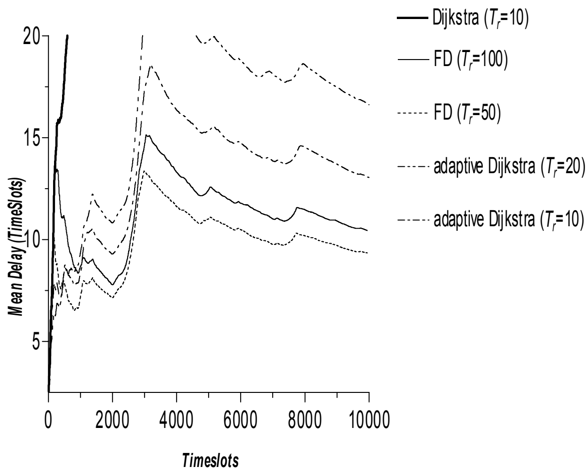

Figure 10. Real-time simulation for the *Self-Similar* case for the FD, *Dijkstra* and adaptive *Dijkstra* algorithms with mean input load 0.4 packets/timeslot.

The results for the case of *Self-Similar* input traffic are dramatically different as we can see in Fig. 10. We observe a high degradation of the system performance even if the mean value of the offered load is equal to 0.4 packets/Timeslot. The *Dijkstra* algorithm presents unacceptable behavior since the mean delay is monotonically increasing until the queues of the network begin to drop packets, but the other two algorithms manage to route the traffic. The *Self-Similar* traffic is characterized by its bursty nature which is the worst event for any routing algorithm, because it may lead to unpredictable peaks. The FD algorithm results to the better value of the mean delay but the stable performance of the system have been lost (observe the shape of the curves compared with the *Poisson* case Fig. 8). The FD algorithm keeps the advantage of the big routing interval T_r=100, compared to the adaptive Dijkstra interval T_r=20. If we decrease the routing interval T_r=50 and T_r=10 for the FD and adaptive *Dijkstra* correspondingly, the mean delay is decreased but the shape of the curve is still the same. The influence is bigger for the adaptive *Dijkstra* algorithm.

In Fig. 11 is studied the *Self-Similar* case for mean traffic density equal to 0.7 packets/timeslot. It is clear that only the FD

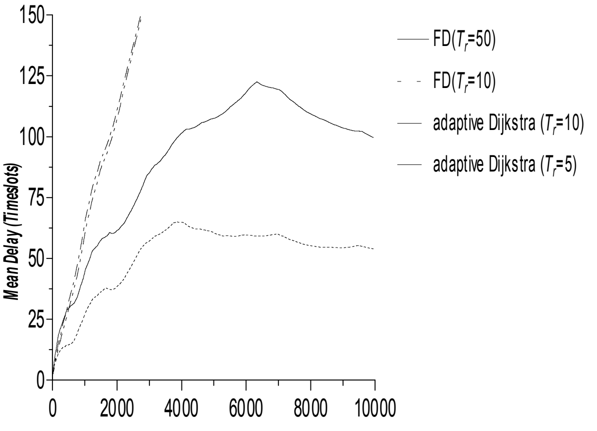

Figure 11. Real-time simulation for the Self-Similar case for the FD and adaptive *Dijkstra* algorithms with mean input load 0.7 packets/timeslot.

algorithm can treat such a heavy load condition with bursty characteristics. The adaptive *Dijkstra* fails even if the frequency of routing triggering is doubled (T_r=5). Generally, the frequency and the policy of routing triggering has been proved to be very important parameter for the design of the system and it is deserved further investigation.

IV. CONCLUSIONS

A comparative study of different routing algorithms has been carried out on LEO satellite constellations. The well used approaches of shortest path algorithms were applied together with the flow deviation of optimal routing techniques. Balanced and unbalanced traffic load and uniform and non uniform distribution of earth stations has been considered and some trials with the link length function have been investigated. Also the *Self-Similar* traffic model is examined versus classical *Poisson* model trying to simulate more realistic conditions of modern networks.

In any case the performance of flow deviation techniques proved to be more reliable. However, due to the complicated topology of the system we proposed and applied successfully a modification of FD algorithm choosing only a limited number of paths to work with. A quantitative estimate of the number of paths has been done through extended simulation running and it was proved that we can find always a very low number of paths to work without loosing in quality of performance. The real-time simulation technique for *Self-Similar* traffic illustrates that classic routing techniques such as shortest path routing algorithms will result to a system degradation since the flow control mechanism of the network will automatically decrease the throughput in order to avoid any congestion periods. On the other hand optimal routing techniques succeed even in heavy traffic conditions to split the traffic among different paths (virtual circuits) and ensure the performance of the system. LEO constellations are very suitable for such routing algorithms because their network architecture presents symmetry and high degree of connectivity.

Some interesting suggestions have been appeared through our study and are worthy for further investigation. So far we have considered only transparent procedures for hand-off assuming zero bandwidth problems. Of course this is not a realistic assumption. Hand-offs load the system with additional delay and a further investigation of the topic is very interesting. Also an interesting point is the study of more realistic cost functions covering propagation delay, hand-off delay, Doppler effects, on-board processing time, etc.

REFERENCES

[1] T. Ors, T. Sammut, L Wood, B. Evans «An overview of Future Satellite Communication Options for Lan Interconnection», COST 253 TD(98)-4.

[2] D. Bertsekas, R.Gallager, *Data Networks* Prentice Hall Inc. 1992, 2nd edition

[3] A. Girard, B. Sanso, «Multicommodity Flow Models, Failure Propagation, and Reliable Loss Network Design», IEEE Trans. On Networking, Vol. 6, No 1, Feb 1998.

[4] E.Papapetrou, I.Gragopoulos, F-N. Pavlidou, "Routing techniques for Low Orbit Satellite Configurations", Proc.of ICT'98, June 21-25, Greece

[5] M.Werner, G.Maral "Traffic Flows and Dynamic Routing in LEO Intersatellite Link Networks" Proc. Of the International Mobile Satellite Conference (IMSC'97) Pasadena , 1997, pp. 283-288

[6] G.Gallo, S.Pallotino, "Shortest Path Algorithms", Annals of Operations Research, No 3, 3-79.

[7] E. W. Dijkstra, «A note on two problema in connections with graphs» Numerische Math, Vol 1, pp. 269-271, 1959

[8] L.Fratta, M.Gerla, L.Kleinrock, "The Flow Deviation Method: An Approach to store-and-forward Communication Network Design", Networks, 1973, No 3, pp.97-133

[9] B. Tsybakov, N. D. Georganas «On Self Similar Traffic in ATM Queues: Definitions, Overflow Probability Bound, and Cell Delay Distribution», IEEE/ACM Transactions on Networking, Vol.5, No.3, June 1997.

[10] I. Gragopoulos, F-N. Pavlidou, "A Study of Routing and Fairness algorithms in Voice-Data Networks", UJOR, 4 (1994), Number 1, 43-66.

[11] E. Papapetrou, F-N. Pavlidou, I. Gragopoulos "Delay and Traffic Load Evaluation of LEO Satellite Constellations with Intersatellite Links under Self-Similar and Poisson Traffic", International Journal of Satellite Communications, accepted for publication, 1999.