

Deflection Routing over Prioritized Intersatellite Links in LEO Satellite Networks

Ömer Korçak and Fatih Alagöz

Boğaziçi University Computer Engineering Department
korcakom@cmpe.boun.edu.tr, alagoz@boun.edu.tr

Abstract

Designing an efficient routing algorithm is a crucial issue in LEO satellite networks for optimizing network resources. Recently proposed priority-based adaptive shortest path routing (PAR) algorithm is a promising technique which was shown to provide high throughput and low delay without any need for signaling overhead. In this work we propose a deflection routing mechanism to be used with PAR. The performance of the proposed algorithm is evaluated for various system parameters through simulations. Simulation results show that the proposed deflection routing approach is promising for low-to-moderate traffic loads, but it fails to improve performance for high traffic loads.

1. Introduction

Due to rapid globalization of telecommunication industry, satellite systems are expected to widely appear in future telecommunication systems, since they provide extensive geographic coverage and wide range of services at competitive infrastructure costs. However, communication over geostationary satellites suffers from high free space attenuation and long propagation delay. Therefore focus has been directed towards development of Low Earth Orbit (LEO) satellite systems with smaller delays. For better utilization of satellites and increasing the performance of the system, new LEO systems usually support on-board processing on satellites and inter-satellite links (ISL) between two satellite nodes. In a LEO satellite network with intersatellite links (ISL), the network layer must decide how to route a packet from a source satellite to a destination. Routing algorithms can be classified as connection-oriented and connectionless algorithms. Connection-oriented algorithms have some advantages like easier handling of QoS guarantees. However, they may suffer in attaining path connectivity by handover mechanisms, since satellite network topology is highly dynamic due to satellite movements [1, 2]. Rather, a distributed next hop routing strategy can be more promising and simpler. That is, considering that each satellite has exact knowledge about the network topology, network layer may select the next hop. To support this, [3] proposes a datagram routing protocol aiming at finding the minimum delay path. It is similar to so called *hot potato* scheme, where a packet is forwarded to the neighboring

satellite that is closest to the exit point. A satellite can change its decision if the ISL to that neighbor is congested. However, we support the idea that it is more appropriate to avoid congestion before it happens, i.e., the algorithm should be proactive. [4] may achieve a better load balancing given that a satellite is aware of the traffic condition at the next hop satellite. In this technique, a congested satellite sends a signal to its neighboring satellites to decrease their sending rates, and its neighbors search for alternative paths. Unfortunately, this technique introduces signaling overhead due to excessive feedback messages. Moreover, this technique does not take any action for load balancing until some nodes experience a level of congestion, i.e. it is not a proactive algorithm either. In [5], we proposed a distributed proactive algorithm, namely Priority-based Adaptive Routing (PAR) algorithm, which aims to balance the traffic before any congestion occurs. We consider that there may be many minimum hop paths between a source-destination pair in a satellite constellation. Therefore, at each satellite, there may be multiple outgoing links that are on one of these minimum hop paths. In the context of PAR, when a satellite node receives a packet, it selects one of these ISLs depending on the priority mechanism. In next section, we provide an overview of PAR algorithm. PAR uses only the ISLs that are on a shortest path. Even in the case that these paths are congested, it does not utilize other links. However, it could be appropriate to utilize a deflection routing mechanism in that case. For this purpose, we propose a deflection routing algorithm in this work, and evaluate it in a simulation environment. Section 3 defines and describes the proposed deflection routing algorithm. We describe the simulation environment in section 4 and present the results in section 5. Finally, section 6 concludes this work.

2. Overview of PAR

Due to highly uniform and symmetric nature of satellite constellations, there may be many minimum hop paths between two nodes in a satellite network. Decision on sending the data over which of those paths has an important effect on the distribution of traffic and utilization of ISLs. PAR aims to set the appropriate path in a distributed manner. When a satellite node receives a packet, it checks whether there are more than one outgoing link which is on a minimum hop path. If so, it selects the one with the highest priority. If the ISL with

the highest priority is congested at that instant, then the ISLs with lower priorities are selected. If all of the ISLs (that are on a minimum hop path) are congested, then packet is dropped. Priorities of links dynamically change depending on the past utilization and queuing information. We introduced a priority mechanism based on the following metric:

$$\mu = \alpha \cdot u_r + \beta \cdot l_q + \delta \cdot d_d \quad (1)$$

where u_r is the link utilization ratio, l_q is the average queue length and d_d is the average dropped data. Each link has its own μ value, and it is updated depending on the changes in the traffic. α , β and δ are design parameters.

Considering that the latest utilization and buffering information is more important than the older ones, we proposed to utilize an aging mechanism while computing the priority metric. We defined an aging period with length t_a . At the beginning of each period, the current μ value is stored in a variable called μ^o . Then satellite starts to collect utilization and buffering information in a new variable called μ^n . At t_0 'th time unit of a given period, μ is calculated as follows:

$$\mu = \mu^o \cdot \left(1 - \frac{t_0}{2t_a}\right) + \mu^n \cdot \left(\frac{t_0}{2t_a}\right) \quad (2)$$

In [5], it is shown via simulations that, PAR is a promising technique for use in LEO satellite networks. PAR uses only the ISLs that are on a minimum hop path. In the case that these paths are congested, it drops the packet and does not utilize other links. However, instead of dropping packets, it could be more appropriate to utilize a deflection routing mechanism. In the following section, we define and describe a deflection routing strategy to use together with the PAR algorithm.

3. Deflection Routing Algorithm

In PAR, each satellite forwards a packet to one of its neighbors that is on a minimum-hop path for the corresponding packet. Now, we define a deflection routing mechanism which will be used when all of the outgoing links towards those neighbors are congested. The proposed deflection routing algorithm is as follows:

When a satellite receives a packet (from a terrestrial node or a satellite node):

- It checks the outgoing ISLs that are included in one of the shortest paths from the source node of the packet to its destination. Let's say these ISLs, *primary ISL*. Among these ISLs, firstly it tries to send the packet from the link with highest priority. If it is congested, it tries other primary ISL(s), if there exists any.
- If all of the links over a minimum-hop path are congested, we select an ISL that is not on a shortest path. The link for deflection must be a neighboring link of one of the primary ISLs, and we call these ISLs, *secondary ISL*. For example, consider a

constellation with 4 ISLs per satellite: West (W), East (E), North (N), South (S). For a particular packet, if N is the only primary ISL, W and E are the secondary ISLs. If N and E are the primary ISLs, W and S are the secondary ISLs. Among the secondary ISLs, decision of which ISL to deflect the packet depends on the same priority mechanism. If the secondary ISL with high priority is congested at that instant, then the one with low priority is selected. If that link is also congested, then packet is dropped.

- In the case of deflection, ID of the corresponding satellite is written over the packet, in order to prevent the packet to revisit that satellite. Otherwise, the routing algorithm will not be loop-free.

Another issue in the context of deflection routing is the threshold for number of deflections. If no threshold is defined, packets may waste resources unnecessarily. Therefore we propose to supply a threshold as follows: When a packet needs to be deflected, we account for the number of hops it has traversed so far. If it exceeds minimum hop distance between the source satellite node and the corresponding node, with a predetermined threshold, packet is dropped. Otherwise it is deflected. To formulate this, we define h_i (number of hops packet p_i traversed so far), h_{xy} (minimum hop distance between satellite x and satellite y) and d (predetermined threshold). If s is the source node of the packet and c is the corresponding node, a packet could be deflected if the following situation holds:

$$\{h_i < h_{sc} \cdot d \mid h_i = 0\} \quad (3)$$

It is clear that none of the satellites (except the source node) support deflection routing at $d = 1$, and packets will always be deflected for large d values, if possible. Here, the question we tackle is "which d value should be set to improve system performance?" We investigate the answer in section 5 for various traffic load characteristics.

4. Simulation Setup

In this work, we consider same simulation environment as in [5]. This section briefly describes considered satellite network topology, traffic model and simulated routing algorithms.

4.1. Satellite Network Model

We consider a polar LEO constellation, similar to Teledesic, with 12 planes and 24 satellites per plane at a height of 700 km. It is a π -constellation, where there is a seam between satellites moving in opposite direction and we assume that there is no ISL passing the seam. Figure 1 shows the considered network topology. For the sake of simplicity, we assume that satellites have disjoint footprints and dividing the earth surface into 12×24 terrestrial zones, each satellite sees one of these zones. Another assumption is made on in the handover mechanism, i.e., as the satellites move with angular

velocity of 3.6 degree per minute, they switch their zones in a discrete manner. This means that their corresponding terrestrial zone changes at each 250 seconds. They complete their rotation in 100 minutes. For simplicity, all ISLs are assumed to be identical (in terms of length and capacity) and their capacity is assumed to be 0.16 Gbps. Each ISL has a buffer of size 40 Mbytes. A packet size is assumed to be 1 Kbytes. Therefore, ISL capacity and buffer size are considered as 20000, and 40000 packets, respectively.

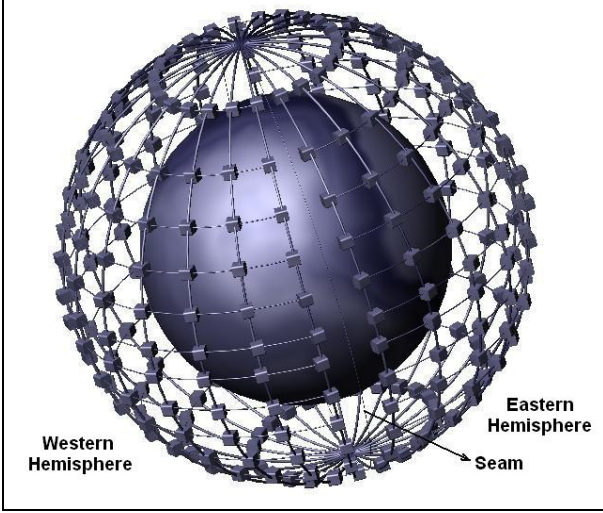


Figure 1. Polar constellation topology with 12x24 nodes

4.2. Traffic Model

As we mentioned above, we divide the earth surface into 12×24 zones. Our traffic model depends on the 2005 statistics about the *user density levels per zone*, *Internet host density levels per continent*, and *user activity levels per hour* (For numerical values of these statistics, please refer to [5]).

Assuming that the host distribution over a continent is distributed proportional with the user density, we calculate host distribution level per zone. Let u_x be the user density level of zone Z_x , and h_x be the host density level. Traffic requirement from zone Z_x to zone Z_y , T_{xy} , is proportional with u_x and h_y , and inversely proportional with distance $dist_{xy}$ between these zones:

$$T_{xy} = \frac{(u_x \cdot h_y)^\theta}{(dist_{xy})^\psi} \quad (4)$$

In the simulations, we set $\theta = 0.5$ and $\psi = 1.5$. Depending on this traffic requirement matrix, we model the traffic. We assume that, at a given hour h , the arrival of a packet with source = Z_x and destination = Z_y is a poisson process with rate $\lambda(x,y,h)$ packets/second:

$$\lambda(x,y,h) = \frac{T_{xy}}{\sum_{\forall i} \sum_{\forall j} T_{ij}} \cdot \frac{a_h}{100} \cdot \frac{A}{3600} \quad (5)$$

where, h is the current local hour and a_h is the activity percentage in the corresponding hour (h). Moreover, A is the aggregate traffic that represents total traffic generated worldwide (packets per day).

4.3. Simulated Routing Algorithms

In our simulations, we basically compare four algorithms:

1. *PAR*: It is the proactive adaptive shortest path routing algorithm that we described in section 2.

2. *Deflection Enabled PAR (DEPAR)*: PAR with deflection routing that is described in section 3.

3. *Fixed Adaptive Routing (FAR)*: FAR is a non-proactive adaptive routing algorithm. For the considered polar constellation topology there are four ISLs per satellite (except the satellites that are in the border of the seam). Two ISLs are on the y direction (North and South) and two are on the x direction (East and West). At each hop, if there are multiple outgoing links that are over a minimum hop path, then FAR selects the one that is on y direction. If that link is congested, then the ISL in x direction is selected. If that link is also congested, then the packet is dropped.

4. *Deflection Enabled FAR (DEFAR)*: FAR with deflection routing. Same deflection algorithm is used as DEPAR. However, among the secondary ISLs, which link to deflect first is decided randomly. If the decided ISL is congested at that instant, then the other secondary ISL is selected. If that link is also congested, then packet is dropped.

For all algorithms, in the case of contention of two packets, we choose the one randomly.

5. Simulation Results

We developed our own simulator in C++. We tested the performance of routing algorithms in terms of drop ratio, average queue length per link and average hop count. *Drop ratio* is defined as the ratio of dropped packets to the sum of dropped and successfully transmitted packets, *average queue length* is the ratio of the sum of the average number of packets in all buffers to the number of ISLs, and *average hop count* is the average number of hops traversed for successfully transmitted packets. While the first metric is a measure of throughput, second and third metrics are measures for delay.

We set the system parameters to the values shown in Table I. Since, a satellite may try to send packet from multiple outgoing links (especially for the deflection routing case), we think that including data drop ratio in priority metric is not fair. Therefore, we set δ to zero. According to the values given in Table II, $\alpha \cdot u_r$ ranges between zero and one, and $\beta \cdot l_q$ ranges between 0 and 2.

TABLE I
SYSTEM PARAMETERS

Total Simulation time	1 day
Warm-up period	60 seconds
Aging period (t)	25 seconds
α	1
β	0.00005
δ	0

Figure 2 shows the drop ratio versus A (in terms of terabit per day). For DEPAR and DEFAR we set the d value to 1.2. It is evident that proactive priority based algorithms (PAR and DEPAR) outperform non-proactive algorithms (FAR and DEFAR). More interesting observation is on the performance difference between deflection routing enabled algorithms (DEPAR and DEFAR) and their pure versions (PAR and FAR). For low traffic loads, deflection routing enabled algorithms perform better. For example, for $A = 200$ Tbps, drop ratio for DEPAR is 1/18 of the drop ratio for the PAR algorithm (although it is not visible in the Figure 2). For $A = 300$ Tbps, this ratio becomes approximately 1/3. As the traffic load increases, difference between drop ratios of DEPAR and PAR reduces. For $A = 800$ Tbps, performances of two algorithms seem to be same in terms of drop ratio. Same relation is also valid for FAR and DEFAR. While for low traffic loads DEFAR outperforms FAR, for high traffic loads the situation is reversed. This is because deflection enabled algorithms postpone dropping of packets and further increase the traffic load in the system. Packets traverse more hops in the system, but because of the high traffic load, number of packets that reach to destination reduces. For low A values, system can tolerate the extra load caused by the deflection mechanism.

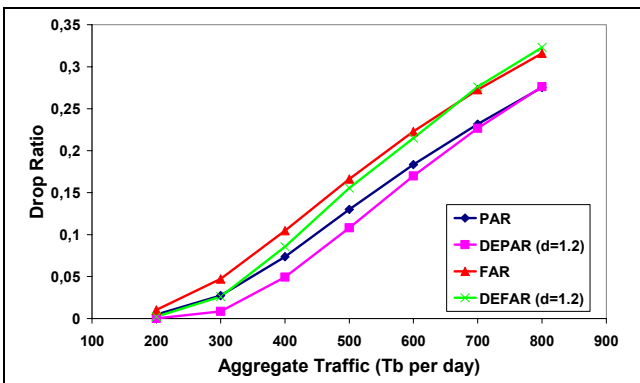


Figure 2. Drop Ratio versus Aggregate Traffic

Figure 3 illustrates the differences between queue lengths for different routing schemes. As the number of successfully transmitted packets increase, we expect that lengths of queues also increase because of the high utilization of links. However, this is not the case for PAR, and it outperforms fixed adaptive routing scheme (FAR) in terms of average queue length. This is because

priority-based techniques provide balanced distribution of traffic among links, and more packets are successfully transmitted with less waiting times in queues. Same comparison is also true between DEPAR and DEFAR. However, according to the obtained results, queue length values for deflection enabled algorithms are worse than that for pure versions. This is because in deflection enabled algorithms, packets stay in the system for longer times. Therefore traffic load is increased, and this results in more waiting times in buffers. However, we ignored retransmissions in our simulations. Algorithms without deflection cause higher packet drops for low aggregate traffic load. This means that higher number of retransmissions is needed. Therefore we can say that, if the retransmissions were taken into account, average queue length values for deflection enabled algorithms and their pure versions would be closer to each other.

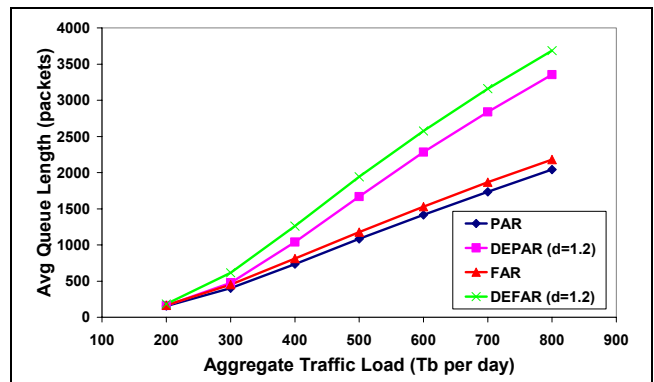


Figure 3. Average Queue Length versus Aggregate Traffic

Next, we examine how deflection routing affects the length of path per packet. According to Figure 4, packets traverse more hops to reach its destination in deflection routing enabled algorithms. This is an expected result because PAR and FAR are shortest path algorithms, whereas deflection routing also utilize longer paths. For PAR and FAR, average hop count decreases as traffic load increases, because packets belonging to long distant routes are exposed to more drops for crowded systems. For deflection enabled algorithms, average hop count per successfully transmitted packets increase with the traffic load up to some point, and then it start to decrease. This could be explained as follows: For very low traffic levels, fewer packets are exposed to deflection. Therefore average hop count is less. As traffic load increases, more packets will be deflected and average hop count increases. However, after a point, crowdedness of the system leads to higher drop ratio for long distance dependent traffic and therefore number of hops traversed per packet starts decreasing.

Obtained results show that for low-to-moderate traffic loads, DEPAR yields better throughput with a reasonable increment in the delay. However, as traffic load increases, pure PAR algorithm becomes more advantageous than DEPAR.

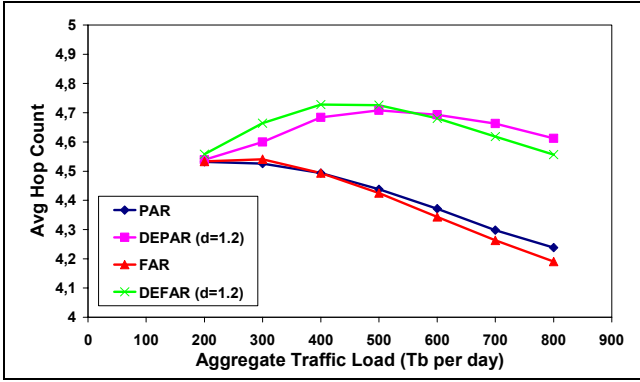


Figure 4. Average Hop Count versus Aggregate Traffic

Until this point we set d value for DEPAR to 1.2. However, the effect of the d value should be investigated for improved system performance (throughput, delay, etc) under various traffic conditions. For this purpose we run our simulations for various d values under various traffic loads.

Figure 5 suggests that drop ratio is not much affected from the d value. We think this is because links near to the high-traffic-generating source nodes are generally more crowded and therefore most of the drops occur in initial hops. Therefore increasing d value has very small contribution on the throughput of the system. In DEPAR with $d=1.0$, deflection is allowed only in the source node and therefore perform worse than cases with slightly more d values.

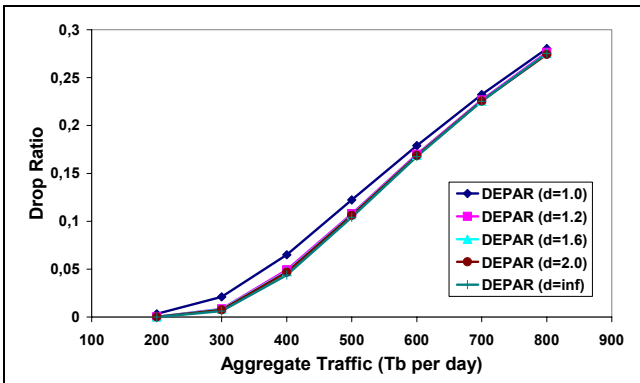


Figure 5. Drop Ratio versus Aggregate Traffic for various d values

Although the throughput is not much affected from the d value, this is not the case for queue length and average hop count traversed per packet. As shown in Figure 6 and Figure 7, there is a perceptible effect of d value on queue length and number of hop traversed. These observations suggest that when we use DEPAR, it is reasonable to keep d value in low levels. However, we should note that these results are obtained for our simulation model (with the given network topology, given traffic model, etc.). Therefore, results may change for different topology and traffic generation characteristics.

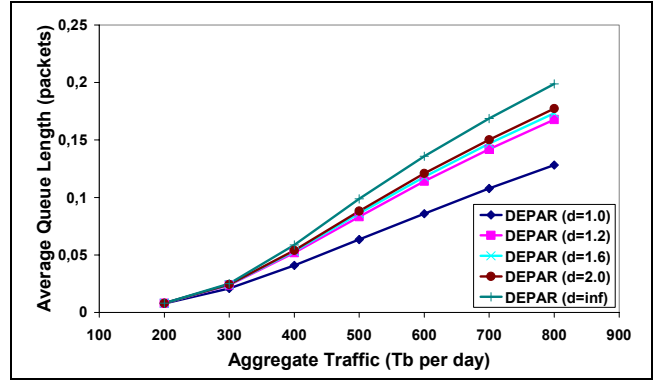


Figure 6. Average Queue Length versus Aggregate Traffic for various d values

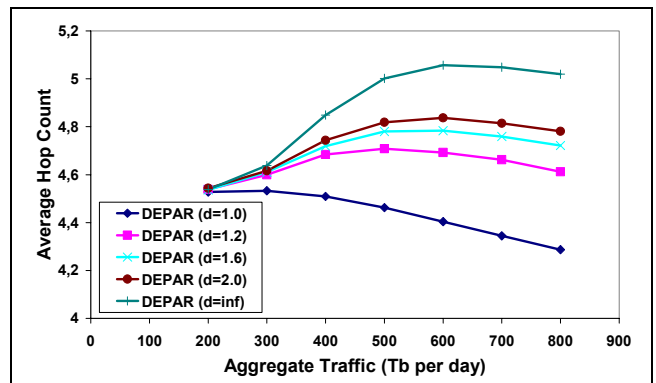


Figure 7. Average Hop Count versus Aggregate Traffic for various d values

6. Conclusions

This paper provides an extension for recently proposed Priority-based Adaptive Routing (PAR) technique. PAR is a distributed adaptive routing technique which uses past utilization and buffering information at links while setting a path between two satellites. PAR does not have any signaling overhead and it was shown to be promising for use in LEO satellite networks. In this work, we propose a deflection routing mechanism which aims to further increase the performance of the system. Simulation results show that proposed deflection routing approach is promising for low traffic loads, but it fails to improve performance for high traffic loads. Including traffic load sensitivity to the deflection mechanism would be an interesting subject of a future study. Moreover, our deflection mechanism could be slightly modified for handling satellite failures.

Acknowledgements

This work is supported by the State Planning Organization of Turkey under “Next Generation Satellite Networks Project”, and Boğaziçi University.

References

- [1] M. Werner, "A Dynamic Routing Concept for ATM based Satellite Personal Communication Networks", *IEEE Journal on Selected Areas in Communications* 1997; **15**(8): 1636-1648.
- [2] H. Uzunalioglu, I.F. Akyildiz, Y. Yesha, W. Yen, "Footprint Handover Rerouting Protocol for LEO Satellite Networks" *ACM-Baltzer J. Wireless Networks (WINET)*, vol.5, no.5, pp.327-337, Nov.1999.
- [3] 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*, 2001; **9**(2): 137-147
- [4] T. Taleb, A. Jamalipour, N. Kato, and Y. Nemoto, "IP Traffic Load Distribution in NGE0 Broadband Satellite Networks", *Proc. of 20th Int. Sym. on Comp. & Inf. Sciences (ISCIS)*, Istanbul, Turkey, Oct. 2005.
- [5] Ö. Korçak, F. Alagöz, "Priority-based Adaptive Shortest Path Routing in IP over LEO Satellite Networks", *23rd AIAA International Communications Satellite Systems Conf. (ICSSC'05)*, Rome, Italy, Sept. 2005.