Multiservice On-Demand Routing in LEO Satellite Networks

S. Karapantazis, E. Papapetrou, *Member, IEEE*, and F.-N. Pavlidou, *Senior Member, IEEE*

***Abstract*—In this paper, a distributed on-demand routing protocol for Low Earth Orbit (LEO) satellite systems, named multiservice on-demand routing (MOR), is proposed and evaluated. The proposed protocol adjusts the routing procedure to the QoS requirements of different traffic classes. The performance of the MOR protocol is compared to the unique proposal for traffic class dependent routing in the literature and the good characteristics of the proposed scheme are corroborated by ample simulation experiments, where significant gains in performance are witnessed.**

***Index Terms*—Routing, on-demand, satellite networks, Low Earth Orbit (LEO), multiservice systems.**

I. Introduction

NOWADAYS we are in the midst of a global revolution in information technology. New requirements for flexible network access have emerged within the telecommunications community, spurred by the vision of optimal connectivity anywhere, anytime. In this context, LEO satellite systems can be instrumental in the future network infrastructure, playing a multifaceted role [1]. In particular, they can be summoned to unify far-flung groups of people, provide reliable wireless access to a unified IP-based core network in the failure of terrestrial infrastructure, realize pervasive access to the Internet and support the development of innovative applications with high bandwidth and low end-to-end delay requirements. Such applications are motivated by the unprecedented growth of the Internet and are expected to thrive with the advent of 4G networks. In light of the above, it becomes evident that the designers of this kind of networks will call for routing protocols that would be able to provide high performance and different QoS levels while keeping complexity and signaling overhead to a minimum at the same time. In polar LEO satellite systems, contiguous satellites can be interconnected through direct links between each other, called Inter-Satellite Links (ISLs). There exist two kinds of ISLs: links between neighboring satellites in the same orbital plane, termed intra-plane ISLs, and links between neighboring satellites in adjacent orbital planes, called inter-plane ISLs. While intra-plane ISLs are permanently maintained, inter-plane ISLs must be broken at the highest latitudes of each orbit due to adverse pointing and tracking conditions as the planes cross and are reestablished as the satellites move to lower latitudes. Thus, in this kind of constellations, implementing multiservice and dynamic routing is a daunting challenge.

The bulk of the studies on this topic capitalize upon the periodic and predictable network topology and divide the orbit's period into several time intervals within which the topology of the system is fixed. For the periodic computation of the shortest path for any pair of satellites, a shortest path algorithm [2]–[4], the Flow Deviation approach [5], or even optimization techniques [6] may be used. Furthermore, some of the studies [3], [5] take account of queuing delay in the estimation of the shortest paths in order to provide adaptability to traffic conditions in the network. Even though the periodic approach offers acceptable implementation complexity since the shortest paths can be computed off-line, this comes at the expense of poor performance since they fail to adapt to frequent variations of traffic related metrics such as queueing delay and link loading. Additional downside includes reduced reliability, in case of link failures, and scalability, since signaling overhead depends on network size and the period of the protocol. To eliminate such disadvantages some researchers have introduced load balancing mechanisms that apportion traffic between two [7], [8] or more different paths [5] or switch a selected path whenever an involved satellite becomes congested [9]. However, the limitation of using periodically computed paths still holds in those studies. To cope with the aforementioned shortcomings some routing schemes that do not implement periodic route calculation have been proposed [10],[11]. The first scheme [10] capitalizes on the system geometry and uses a congestion avoidance mechanism, while the second [11] employs an on-demand path discovery process.

Despite the numerous studies, most of the schemes do not provide service differentiation although it is indisputable that in the future, LEO satellite networks will be called to support a bunch of applications in order to be viable. Only the study in [12], [13] dealt with the issue of traffic class dependent routing. In particular, periodic path calculation based on the Dijkstra algorithm is employed along with a link-cost metric that consisted of the sum of the propagation and queuing delays for both delay-sensitive and best-effort traffic, whereas for throughput-sensitive traffic the normalized available bandwidth of the link was considered as the link-cost metric and the Bellman-Ford shortest path algorithm was adopted to compute paths that offer the maximum available bandwidth within a minimum hop count. The major drawback of the scheme is that it depends on the periodic scheme for path calculation and, therefore, inherits all its limitations. This observation has been the main drive behind our work. In this paper we capitalize upon the deterministic variations in the topology of LEO satellite systems and propose a topology-agile protocol, called (MOR). MOR can be viewed as an extension of the

Manuscript received March 8, 2008; revised July 10, 2008; accepted September 4, 2008. The associate editor coordinating the review of this letter and approving it for publication was W. Zhuang.

S. Karapantazis and F.-N. Pavlidou are with the Telecommunications Division, Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece (e-mail: {skarap, niovi}@auth.gr).

E. Papapetrou is with the Department of Computer Science, University of Ioannina. 45110, Ioannina, Greece (e-mail: epap@cs.uoi.gr).

Digital Object Identifier 10.1109/TWC.2009.080334

LAOR protocol [11] for multiservice LEO networks. The prime aim of MOR is to take advantage of the *on-demand path computation* to collect up-to-date link measurements and compute the optimum path. It is well-known that the signaling of on-demand schemes aggravates with the number of connections served by the network. Although this number is expected to be high in the context of global broadband LEO satellite networks, the normalized signaling is expected to be low considering that connections will be broadband. Furthermore, MOR inherits from LAOR its ability to *minimize the overhead involved in discovering a path*. Finally, contrary to LAOR, MOR provides service differentiation by using a modified route computation mechanism and different cost metrics for each traffic class.

II. THE MOR PROTOCOL

For the description of the protocol the virtual topology that was proposed in [11] will be employed, according to which the network can be modeled as a graph $G(V, E)$, comprising a set of nodes V and a set of edges E (fig. 1). In this graph each satellite is uniquely defined by the pair of virtual coordinates $(\overline{x}, \overline{y})$, where $\overline{x}$ and $\overline{y}$ denote the orbital plane and the position of the satellite in this plane respectively. In addition, the network is considered to support three different service classes [12], [13]:

- **Class A:** typical real-time services such as Voice over IP (VoIP) and interactive video applications. This traffic class is sensitive to packet delay, which needs to be minimized.
- **Class B:** best-effort traffic, which has no specific QoS constraints.
- **Class C:** bandwidth-sensitive services such as video on-demand and distribution of large files. This traffic class requires throughput to be maximized.

The operation of the MOR protocol can be broken down into two procedures: i) *route request area formation*, and ii) *path discovery*. The latter is the cornerstone of MOR and involves the exchange of *route request* (RREQ) and *route reply* (RREP) packets between the satellites in order to obtain the optimal path. However, in order to minimize the signaling overhead, the area where RREQs will be sent out is limited to the *route request area* (fig.1). This area is determined by the *route request area formation* procedure which is inherited by LAOR [11] and is used for all traffic classes. The boundaries of this area are defined by the originating and destination satellites, as shown in fig.1, and can be expanded by a parameter $width$ in order to allow for more alternative paths without increasing routing overhead. Experiments conducted in [11] revealed that the best value for $width$ depends on the positions of the originating and destination satellites as well as on the design of the satellite network. The methodology and related equations for the calculation of parameter $width$ can be found in [11].

Nevertheless, unlike LAOR, MOR provides service differentiation by employing different cost metrics for each traffic class. Consequently, different paths may result for different traffic classes, which translates in that each satellite maintains three independent routes for each origin/destination pair,

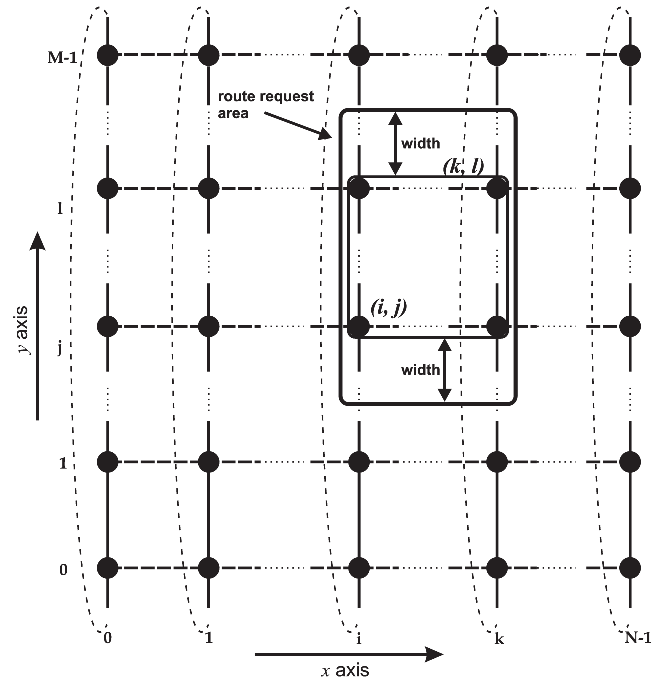

Fig. 1. Virtual topology for polar LEO satellite networks.

corresponding to each traffic class. Morever, MOR does not hinge on the concept of sequence numbers. Instead, it relies on the clock that each satellite maintains to uniquely mark the RREQ and RREP packets that are exchanged during the path discovery phase in order to compute the optimal path between two satellites. Last but not least, it will be explained in Section II-C that the *path discovery* process is modified compared to that of LAOR in order to support both additive and concave cost metrics.

A. Packet and Routing Table structure

Before delineating the *path discovery* procedure, let us cast light to the packets as well as the routing tables that MOR makes use of. The RREQ packet consists of seven fields, whereas the RREP comprises eight fields. The following six fields are included in both types of packets. The field *type_id* determines whether the packet is a RREQ or a RREP, whilst the field *class_id* is used to indicate the target traffic class of the path discovery process. The *src* and *dst* fields are used to indicate the originating and the destination satellites of the packet. The cost of the formed path is stored in the field *path_cost*, while the *exp_time* field contains the time instant at which the path will become invalid due to the switching off of an ISL. In the case of a RREQ packet, both of the aforementioned fields contain values that refer to the path from the source satellite to the satellite that the RREQ packet has reached. On the other hand, in a RREP packet the values of the same fields refer to the whole path from source to destination, namely the values contained in the RREQ packet when it reached the destination. In addition to these fields RREQ packets also have another field that contains the time instant that the packet was sent out by the originating satellite, i.e., the *timestamp* field. On the other hand, in a RREP packet there exist two fields, namely *src_timestamp* and *dst_timestamp*.

The first one contains the timestamp of the RREQ packet that produced the RREP, while the second contains the time instant that the RREP was created and sent out.

Each satellite maintains a routing table RT which contains one entry per destination satellite and traffic class. The entry contains the next hop to the specified destination for the specified traffic class. It also contains the number of hops to the destination, the total cost of the path and the time instant that the path entry will expire (*exp_time*). Moreover, the entry is augmented with three more fields, namely *rt_owner*, *src_timestamp* and *dst_timestamp* that are used to determine whether the satellite is allowed to use this entry to reply to requests. The field *rt_owner* is boolean and is set to TRUE only if the route to *dst* was discovered as a result of a path discovery process initiated by this satellite. Each satellite can reply to route requests only with routes that it owns. Furthermore, the timestamps are used to avoid loop formation (*src_timestamp*) and ensure the freshness of cached routes (*dst_timestamp*). Additionaly, each satellite maintains a routing table $ReqT$ used to store reverse routes for delivering RREP packets back to the satellite that originated the RREQ packet. The need for the $ReqT$ table lies in the fact that for a given pair of satellites the optimal forward and reverse paths in general are not identical since the traffic that flows in both directions is not the same. Therefore, a RREQ packet can set up a path only toward the destination satellite. However, a path toward the originating satellite is also needed in case a RREP packet must be sent to it. In particular, the entries of a $ReqT$ contain the timestamp (*orig_timestamp*) of the RREQ that produced the entry, the originator of the RREQ packet (*orig*), the next hop to the originator of the request, the cost of the path toward the originator, as well as the number of hops of the path from the originator of the RREQ to this satellite.

B. Path Discovery

MOR is executed in each satellite and is invoked whenever a ground terminal initiates a connection with another terminal or when an active terminal is handed over to another satellite. In order to capture the current network conditions the algorithm is initiated even if a route to the destination satellite exists. All the packets that arrive to the satellite until a path is set up are temporarily stored in a queue. Alternatively, the flow of data may begin after the path setup in order not to burden the satellite with the task of queuing packets. The *path discovery* process commences as soon as the *route request area* is formed. The originating satellite generates a RREQ packet which will be delivered only to the neighboring satellites that belong to the *route request area*. When an intermediate satellite int with virtual coordinates (x_{int}, y_{int}) receives a RREQ packet, denoted by $rreq$, first checks if its RT table for the respective traffic class contains an entry for the destination satellite. Supposing that there does not exist such an entry, the intermediate satellite has to decide whether the RREQ is fresh enough to forward it or not. To this end, it will dig into its $ReqT$ table to see if an entry for the RREQ's originator exists. The intermediate satellite cancels the forwarding of the RREQ only if the $ReqT$ table contains an entry rve for the RREQ's originating satellite and $rreq.src\_timestamp < rve.orig\_timestamp$ or if these two fields are equal and $rreq$ has arrived through a path with greater cost than the RREQ packet that generated the entry rve. The aforementioned criteria are based on the concept that there is no need to forward a RREQ packet that is not fresher than the last known RREQ from the same originator or has arrived through a longer path. By using these criteria, MOR avoids loop formation and at the same time minimizes the involved overhead. In any other case, the intermediate satellite, after appropriately updating the RREQ's fields, will forward it toward the destination satellite (x_{dst}, y_{dst}). That means that if $x_{dst} \geq x_{int}$, then the RREQ will be sent to the satellites that belong to the *route request area* and whose $x-$coordinate is greater than or equal to x_{int}.

Let us now consider the case that the intermediate satellite has an entry re in the relative RT table for the destination satellite. A route entry is considered valid as long as it meets the following preconditions: 1) $re.rt\_owner == TRUE$; 2) $re.src\_timestamp \geq rreq.timestamp$; and 3) the time interval left until the expiration of the path is greater than exp_thd, where exp_thd is a threshold that determines the minimum acceptable lifetime of a newly discovered path. The first condition is used to allow only intermediate nodes that have up-to-date routes to reply to a RREQ, while the second condition guarantees the formation of loop-free routes. Last but not least, the third prerequisite aims to avoid the establishment of paths that will become invalid very shortly due to the switching off of an ISL. The choice of exp_thd should allow enough time for delivering the in-flight packets and setting up a new path in order to avoid packet drops. Considering that an acceptable worst-case end-to-end delay in such systems is below 250 ms or less, an appropriate value for exp_thd could be of the same order of magnitude. Greater values could only increase the routing overhead without affecting the protocol effectiveness. If the intermediate satellite satisfies the aforementioned preconditions, it updates (or creates if not existent) the corresponding entry of table $ReqT$ with the respective fields of the RREQ packet. Then it generates a RREP packet and sends it toward the originating satellite via the path established by the RREQ, namely, using the ancillary $ReqT$ table. The same steps are taken when the RREQ is received by the destination satellite.

C. Cost metrics and Impact on Path Discovery

As mentioned earlier, the most important characteristic of MOR is that it provides service differentiation. Therefore, MOR uses different cost metric for each traffic class with the aim of meeting its QoS constraints. As far as traffic class A is concerned, which represents real-time traffic, the minimization of end-to-end delay is of utmost importance. Thus, the MOR protocol employs a link cost metric that is the sum of the propagation and queuing delays. It must be noted that the cost metric used for traffic class A is additive. This means that each time that a RREQ packet is forwarded by a satellite the recorded cost is updated by adding the cost of the previous ISL. Moreover, when a node replies with a stored route, the cost carried in the RREP packet is the sum of the cost carried in the RREQ packet and the cost of the stored route. Using the

TABLE I
SIMULATION PARAMETERS

♯ of orbits	6	♯ satellites per plane	11
Satellite altitude	780 km	Inclination	86.4^o
Interplane separation	31.6^o	Min. elevation angle	8.2^o
Cross-seam ISLs	No	♯ of ISLs	2intra+2inter plane
ISL latitude threshold	$\pm$ 60^o	Up/downlink bandwidth	15 Mb/s
ISL bandwidth	10 Mb/s	ISL LL queue size	500 packets
Simulation duration	6200 sec	exp_thd	0.5 sec
"On"period	0.3 sec	"Off"period	0.9 sec
Packet length	1500 bytes	Class A Bitrate during "On"periods	$50\% BR_t$
Class B Bitrate during "On"periods	$20\% BR_t$	Class C Bitrate during "On"periods	$30\% BR_t$

sum of propagation and queueing delay as a metric provides implementation simplicity since only the first copy of a RREQ packet needs to be forwarded by a satellite. This is because the first copy always corresponds to the path with the minimum cost. Furthermore, a satellite is not required to perform any measurements to calculate the cost of a link. It relies only on the difference between the time instants of the RREQ packet's transmission and reception. The same cost metric is also used for traffic class B, which represents best effort traffic. The rationale behind this choice is that this traffic class has loose QoS requirements and therefore, the most dominant criterion for the selection of the most appropriate cost metric is implementation simplicity. Concerning traffic class C, which represents bandwidth-sensitive traffic, the most important parameter is the available bandwidth in each ISL, since this class requires the maximization of throughput. Therefore, the MOR protocol uses a cost metric AB that reflects the available bandwidth of a link. This cost metric is given by $AB = 1 - ISL_{util}^k$ and ISL_{util}^k is equal to:

$$ISL_{util}^k = link\_state + decay \cdot \left(ISL_{util}^{k-1} - link\_state \right) \quad (1)$$

where ISL_{util}^k and ISL_{util}^{k-1} denote link utilization when packets $k-1$ and k arrived in the interface queue respectively. The parameter *link state*, which is provided by the link layer (LL), is set to 0 if there is no packet either in the interface queue or under transmission, otherwise it is set to 1. The parameter $decay$ denotes the *forgetting factor* [8] and is given by $e^{-\Delta t_k}$, where Δt_k is the time interval between the arrivals of packets k and $k-1$ in the queue. It should be mentioned that this cost metric is concave. Thus, each time that a RREQ packet traverses an ISL, the satellite compares the ISLs cost AB to the value of the RREQ's corresponding field and updates this field with the minimum of these values. Similarly, when an intermediate node replies to a RREQ the total cost AB of the path is the minimum of the cost carried in the RREQ and the cost of the stored route. Using a concave metric significantly modifies path discovery since intermediate satellites may need to forward more than one copies of a RREQ since a later copy may represent a less congested or shorter path. Similarly, a destination node may need to produce more than one RREP packet. This represents a significant departure from the discovery process used in LAOR. In addition to the cost metric AB, in order to keep end-to-end delay to low levels and avoid wasting resources by spreading traffic over the satellite network, the number of hops is used as a second metric for traffic class C. Therefore, among all the possible paths that connect two satellites the path that consists of the minimum number of hops and the maximum available bandwidth is opted. The combined use of two metrics to calculate the optimal paths on an on-demand basis is another innovation introduced by MOR in the path discovery process.

III. SIMULATION RESULTS AND DISCUSSION

For the performance evaluation of MOR the existing satellite component of the network simulator (ns-2) has been expanded. MOR was compared to the sole proposal as yet for multiservice LEO networks, i.e., the Traffic Class Dependent Routing (TCDR) protocol [12], [13]. In order to ameliorate the performance of TCDR, the period for path calculation T_p was set to 10sec. Furthermore, in order to have a fair comparison, it was considered that in the case of TCDR the delay measurements were available without delay. Similarly, the computed paths were considered to be available at each satellite thereupon their calculation. The MOR and TCDR protocols were assessed using an Iridium-like constellation. For the sake of fairness, it should be noted that although MOR is optimized for polar systems, TCDR is optimized for inclined systems where constant satellite connectivity is assumed. Therefore, in order to minimize the impact of switching on/off of ISL's we also used updates triggered by topology changes. The simulation parameters are presented in Table I. The formation of the route request area was performed according to the methodology in [11]. Traffic was generated by 200 terminals that were distributed over the globe according to the *hot spot* scenario [8], each producing a total of BR_t kb/s.

Figure 2 depicts the average end-to-end delay versus overall terminal's bitrate. It is apparent that MOR outperforms TCDR in regard to any traffic class. In particular, MOR attains very low end-to-end delay even for high bitrates. On the contrary, the performance of TCDR significantly aggravates as bitrate increases. Moreover, while in the case of MOR end-to-end delay for traffic class C packets is greater than the one of packets of traffic classes A and B, as expected, the exact opposite phenomenon is observed in the case of TCDR. The TCDR algorithm periodically calculates the shortest paths for all traffic classes and uses the same paths for the period duration. Furthermore, the paths computed for traffic classes A and B in many cases, and in particular for low and moderate traffic load, happen to be identical. As a result, packets of these traffic classes are frequently transferred to their destination through identical paths, thereby rendering them heavily loaded. Figure 3 illustrates packet delivery ratio versus terminal's bitrate. In

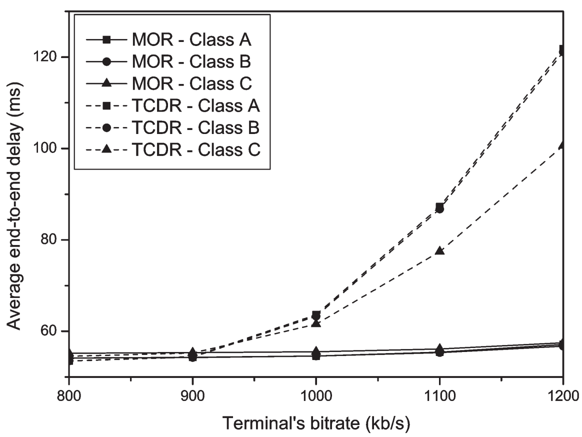

Fig. 2. Average end-to-end delay vs terminal's bitrate.

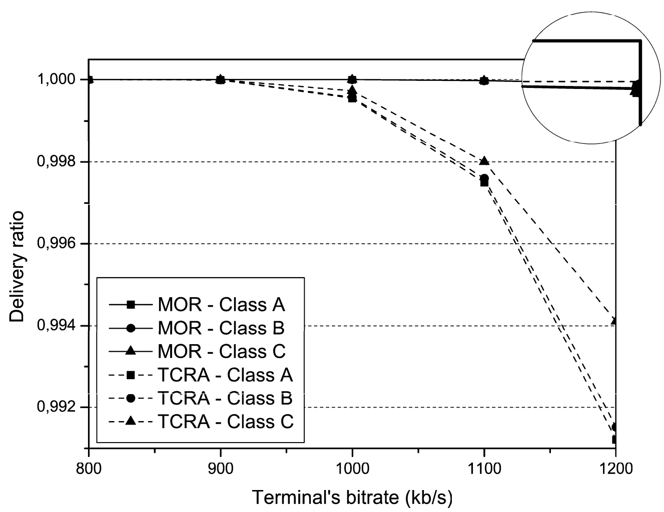

Fig. 3. Delivery ratio vs terminal's bitrate.

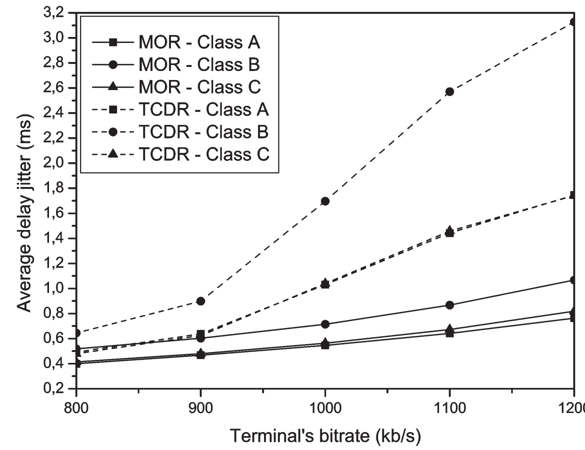

Fig. 4. Average delay jitter vs terminal's bitrate.

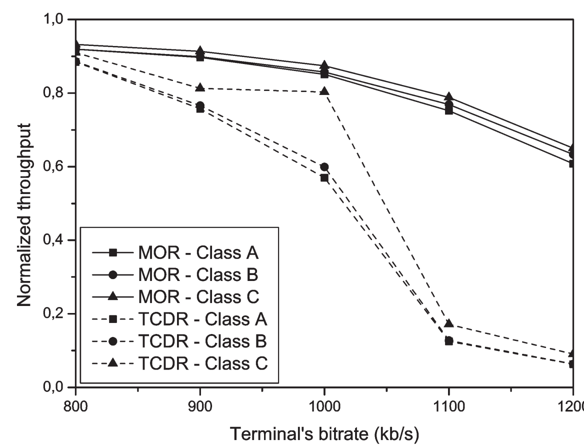

Fig. 5. Normalized throughput vs terminal's bitrate.

general, delivery ratio decreases as traffic intensity increases since packets are dropped due to congestion which results in queue overflow. Nevertheless, MOR minimizes congestion by achieving a better distribution of the traffic over the satellite network and therefore manages to successfully deliver packets of all traffic classes to their destinations even for high traffic load. The fact that MOR achieves better distribution of traffic is also evident in the end-to-end delay as well as in the average delay jitter, presented in fig.4. This figure reveals that by producing a more balanced loading of ISLs, MOR minimizes traffic variations and consequently suppresses delay jitter. This fact proves that MOR is advantageous to real time applications, such as VoIP, where delay jitter is of utmost importance. As far as TCDR is concerned, its delivery ratio decreases as traffic load increases, even in the case of traffic class C due to the increasing queue size in ISLs. Delivery ratio of around 99% is deemed low for some applications that belong to traffic classes A and C, such as VoIP and file transfer.

Figure 5 presents normalized throughput versus terminal's bitrate. Normalized throughput is defined as DT/C [12], [13] where the parameter DT denotes the data throughput calculated as the quotient of the packets' length and the average queuing delay. Namely, for each traffic class, DT is calculated as follows:

$$DT = \frac{N_{samples} \cdot L_{packet}}{\sum QD} \quad (2)$$

where $N_{samples}$ is the number of packets of this traffic class that have been transmitted thus far and $\sum QD$ is the sum of the corresponding queuing delays. The normalized throughput allows us to gage the protocols' ability to satisfy the QoS constraints of traffic class C. Obviously, traffic class C exhibits the highest throughput, as expected. The performance disparities between the two protocols are significant and MOR outperforms TCDR with regard to all traffic classes. It is observed that normalized throughput reduces as bitrate increases on account of increased queuing delay (eq.(2)) due to packet congestion in ISL queues. This is also confirmed by fig.2. However, the decrease in the case of MOR is much smaller, which again proves that MOR manages to efficiently apportion traffic among different paths. Last but not least, we also evaluated the performance of MOR in terms of

TABLE II
NORMALIZED PACKET AND BYTE OVERHEAD (IN %)

	Packet Overhead			Byte Overhead		
Bitrate	Class A	Class B	Class C	Class A	Class B	Class C
800 kb/s	0,519%	1,270%	1,023%	0,016%	0,040%	0,032%
900 kb/s	0,463%	1,136%	0,907%	0,014%	0,036%	0,028%
1000 kb/s	0,417%	1,025%	0,816%	0,013%	0,032%	0,026%
1100 kb/s	0,380%	0,935%	0,740%	0,012%	0,029%	0,023%
1200 kb/s	0,348%	0,860%	0,677%	0,011%	0,027%	0,021%

signaling overhead. Two different performance indicators were employed. The first one is normalized packet overhead, that is, the ratio of the number of routing packets transmitted to the number of data packets delivered, and the second one is normalized byte overhead, namely, the ratio of the total number of bytes related to routing packets to the total number of bytes of delivered data packets. Table II provides a tabulation of the results. In order to account for the worst case, the size of each RREQ's and RREP's field was set to four bytes, although a smaller size would be sufficient for many of these fields (i.e., $type\_id$, $class\_id$, etc). From these tables it becomes evident that signaling overhead can be considered negligible. In addition, byte overhead is much smaller than packet overhead owing to the shorter length of RREQ and RREP packets compared to the length of data packets.

IV. CONCLUSIONS

In this paper the issue of on-demand routing in multiservice LEO satellite systems has been addressed. In particular, the multiservice on-demand routing (MOR) protocol has been proposed which computes independently the optimal paths for each traffic class, aiming at meeting the QoS requirements of each class. Furthermore, it overcomes the shortcoming of on-demand routing protocols, i.e., high signaling overhead, by utilizing the deterministic dynamics of satellite movement. Ample simulation results provided corroboration for the advantages of the proposed protocol, proving that MOR manages to fulfill the QoS requirements of all traffic classes.

REFERENCES

[1] T. Taleb, N. Kato, and Y. Nemoto, "Recent trends in IP/NGEO satellite communication systems: transport, routing, and mobility management concerns," *IEEE Wireless Commun. Mag.*, vol. 12, no. 5, pp. 63-69, Oct. 2005.

[2] M. Werner, C. Delucchi, H. Vögel, G. Maral, and J.-J. D. Ridder, "ATM-based routing in LEO/MEO satellite networks with intersatellite links," *IEEE J. Select. Areas Commun.*, vol. 15, no. 1, pp. 69-82, Jan. 1997.

[3] M. Mohorčič, M. Werner, A. Švigelj, and G. Kandus, "Adaptive routing for packet-oriented intersatellite link networks: performance in various traffic scenarios," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 808-818, Oct. 2002.

[4] H. Uzunalioğlu, I. F. Akyildiz, and M. D. Bender, "A routing algorithm for connection-oriented Low Earth Orbit (LEO) satellite networks with dynamic connectivity," *Wireless Networks*, vol. 6, no. 111, pp. 181-190, 2000.

[5] E. Papapetrou and F.-N. Pavlidou, "A proposal of optimal routing techniques for non-GEO satellite systems," *International J. Wireless Inform. Networks*, vol. 8, no. 2, pp. 75-83, Apr. 2001.

[6] H. S. Chang, B. W. Kim, C. G. Lee, S. L. Min, Y. Choi, H. S. Yang, D. N. Kim, and C. S. Kim, "FSA-based link assignment and routing in low-earth orbit satellite networks," *IEEE Trans. Veh. Technol.*, vol. 47, no. 3, pp. 1037-1048, Aug. 1998.

[7] M. Mohorčič, M. Werner, A. Švigelj, and G. Kandus, "Alternate link routing for traffic engineering in packet-oriented ISL networks," *International J. Satellite Commun.*, vol. 19, no. 2, pp. 463-480, Apr. 2001.

[8] M. Mohorčič, A. Švigelj, G. Kandus, Y. F. Hu, and R. E. Sheriff, "Demographically weighted traffic flow models for adaptive routing in packet-switched non-geostationary satellite meshed networks," *Comput. Networks*, vol. 43, no. 2, pp. 113-131, Oct. 2003.

[9] T. Taleb, A. Jamalipour, K. Hashimoto, Y. Nemoto, and N. Kato, "ELB: an explicit load balancing routing protocol for multi-hop NGEO satellite constellations," in *Proc. GLOBECOM 2006*, Nov. 2006.

[10] E. Ekici, I. F. Akyildiz, and M. D. Bender, "A distributed routing algorithm for datagram traffic in LEO satellite networks," *IEEE/ACM Trans. Networking*, vol. 9, no. 2, pp. 137-147, Apr. 2001.

[11] E. Papapetrou, S. Karapantazis, and F. N. Pavlidou, "Distributed on-demand routing for leo satellite systems," *Comp. Networks*, vol. 51, no. 15, pp. 4356-4376, 2007.

[12] A. Švigelj, M. Mohorčič, G. Kandus, A. Kos, M. Pustišek, and J. Bešter, "Routing in ISL networks considering empirical IP Traffic," *IEEE J. Select. Areas Commun.*, vol. 22, no. 2, pp. 261-272, Feb. 2004.

[13] M. Mohorčič, A. Švigelj, and G. Kandus, "Traffic class dependent routing in ISL networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, no. 4, pp. 1160-1172, 2004.