# A Predictive QoS Routing Scheme for Broadband Low Earth Orbit Satellite Networks

**Özgür Erçetin*+, Srikanth Krishnamurthy$, Son Dao$, and Leandros Tassiulas+**

$Information Sciences Laboratory
HRL Laboratories, LLC,
Malibu, CA 90265.
*{krish,skdao}@hrl.com*

+Electrical and Computer Engineering Department,
and Institute for Systems Research,
University of Maryland,
College Park, MD 20742
*{ercetin,leandros}@Glue.umd.edu*

## Abstract

**Low Earth Orbit Satellite Networks can augment terrestrial wireless networks to provide global broadband services to users regardless of the users' locations. Delivering QoS guarantees to the users of LEO satellite networks is complicated since footprints of LEO satellites move as the satellites traverse their orbits, and thus, causing frequent user handovers between the satellites. Traffic on inter-satellite links of a particular satellite change as the user traffic served by the satellite changes with the satellite's mobility. The change in user traffic on the inter-satellite links may cause violation of QoS requirements of on-going calls. We propose a novel routing algorithm called Predictive Routing Protocol (PRP), that exploits the predictive nature of the LEO satellite topology to maximize the total number of users served by the system, while maintaining each user's QoS requirements. PRP predicts the user traffic load on the inter-satellite links up to a short time in the future by using the deterministic knowledge of the LEO satellite topology, and user location information. PRP determines multiple paths for a particular connection that effectively help avoid possible future bottlenecks as predicted by estimated future traffic on the inter-satellite links. The algorithm is compared with other non-predictive routing protocols such as IP routing by extensive simulations and it is shown that PRP can deliver deterministic QoS guarantees (such as delay jitter), without over-reserving channel bandwidth. An admission control curve has also been obtained which may be used to ensure that the desired QoS metrics may be guaranteed.**

## 1. Introduction

Terrestrial wireless networks (cellular and PCS networks) provide mobile communications services with limited geographic coverage. The *L*ow *E*arth *O*rbit (*LEO*) satellite networks can augment these networks to provide global coverage to a more diverse user population. The round trip propagation delay for communication with a LEO satellite (from an Earth terminal) is comparable to the round trip communication time in terrestrial networks due to the low altitude of the satellites. Real-time communications services can be provided to the users regardless of the users' geographical location. A Low Earth Orbit is any earth orbit of up to approximately 1,500 kilometers in altitude (Figure 1). At this altitude, satellites orbit the earth in approximately 100-120 minutes. Due to attenuation and terrain shadowing effects, reliable communication is not possible at low elevation angles [1]. The low altitude of the satellites and the need for high elevation angles for successful communications necessitate small satellite footprints. In order to provide continuous and seamless services to users regardless of where a particular user is located, LEO satellite networks will have satellite constellations with tens of satellites. These satellites will be equipped with sophisticated technologies such as on-board processing and inter-satellite links and are expected to provide the framework for robust and efficient universal communications.

LEO satellite systems currently under various phases of deployment maintain either Earth-fixed cells (such as the Teledesic satellite network [2]) or they maintain satellite-fixed cells (Figure 2). Earth-fixed cells are cells wherein stationary cells on the earth are dynamically served by LEO satellites moving to within the range of the cell. On the other hand, satellite-fixed cells refer to dynamically moving satellite footprints and the affiliation of an individual user changes from cell to cell in time. The issues in having Earth-fixed cells are similar to the issues in building terrestrial cellular networks. In satellite networks with Earth-fixed cells, the mobility of the terrestrial users rather than the mobility of the satellites cause the hand overs. However, in LEO satellite systems with satellite-fixed cells, due to the movement of the satellite footprint, the number of users in a cell and the traffic served by each satellite changes in time. A user is *handed over* from one satellite to another multiple times during the lifetime of a call. The inherent mobility of the satellites may cause problems in maintaining the user connections. An on-going call may be dropped during handoff, due to the non-availability of a user-to-satellite uplink-downlink channel. If

* This work was done when the author was with HRL Laboratories, LLC.

0-7803-6465-5/00 $10.00 © 2000 IEEE

the connection has strict *Q*uality *o*f *S*ervice (*QoS*) requirements, such as delay or delay jitter bounds, it may be blocked even if user-to-satellite channels are available due to the lack of a route with adequate resources from the satellite entry to the satellite egress point. Provision of Guaranteed Service relies on the reservation of a specific amount of bandwidth for each call on the links connecting the communicating end-users [10]. For example, in terrestrial broadband networks, a route for a particular connection between two end-users is determined based on available bandwidth on various network links at the time of call set-up. This particular route is used for the entire call duration. In LEO satellite networks the traffic on the inter-satellite links (ISLs) also change with changes in the user-to-satellite traffic (which in turn changes due to the mobility of the satellites). Hence, traditional terrestrial routing protocols cannot be applied to broadband LEO satellite networks. Although sufficient bandwidth may be available on a particular route at call set-up for a particular call, the same route may become congested in time due to the changes in access traffic loads which in turn are changing due to the mobility of the satellites.

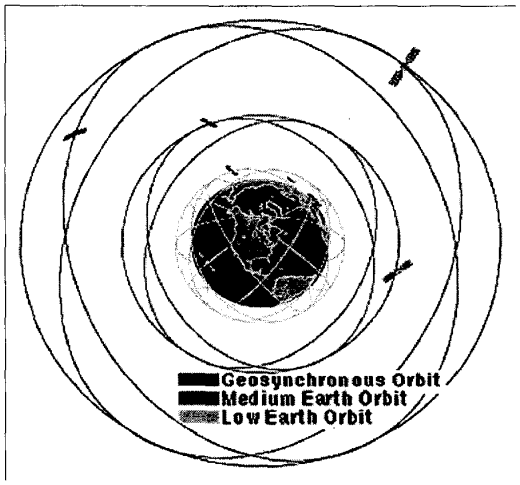


**Figure 1-** *Low, Medium and Geosynchronous Earth Orbits.*

The focus in research in LEO satellite networks has been in providing successful handover to users as they transition from one satellite's coverage area to the coverage area of another. In [3] an analytical model has been proposed for modeling handovers of the users between satellites that are in the same orbital plane. However only uplink access has been considered (single hop scenario) and rerouting on the LEO constellation, which might be needed due to these handovers, has been ignored. In [4] handovers between the satellites in adjacent orbital planes are also considered for a single hop scenario. However, multi-hop communications is necessary in mobile satellite networks since different users might be covered by different satellites. The multi-hop satellite routing problem has been addressed in [5] with an emphasis on setting up routes between pairs of satellites to minimize the re-routing frequency. Notice that, the need for rerouting arises from the fact that, often, no user pair can be serviced by the same satellite end nodes for the complete call duration. In [5] route optimization was performed for the routes between two satellites. Realistically, the optimization is needed for the route between two ground terminals. An optimal route between two satellite nodes is not necessarily optimum for a connection between two ground terminals, since the handovers between the ground terminals and the satellites result in changing satellite end nodes for the connection. The handover rerouting problem has been addressed in the context of terrestrial wireless networks [6, 7, 8]. In terrestrial cellular networks, the cells and the base stations serving those cells are stationary. The handover rerouting problem, then, arises due to the mobility of the end-users rather than the mobility of the base stations. One proposed solution to the handover rerouting problem in terrestrial networks was to determine a whole new route after a handover [7]. This solution, although optimal for the particular connection, causes excessive signaling in the network resulting in a degradation of network throughput. Partial re-routing schemes have also been proposed, wherein the processing and messaging overhead in the network is reduced by chossing a non-optimal path for the connection. Uzunalioglu et al. [9] investigated a simple handover re-routing algorithm called the Footprint Handover Re-route Protocol (FHRP), which finds a new path when a connection is handed over, by using as much of the original path used by the connection as possible. This algorithm reduces messaging and processing overhead considerably, but does not find the optimal path from a source to a destination.

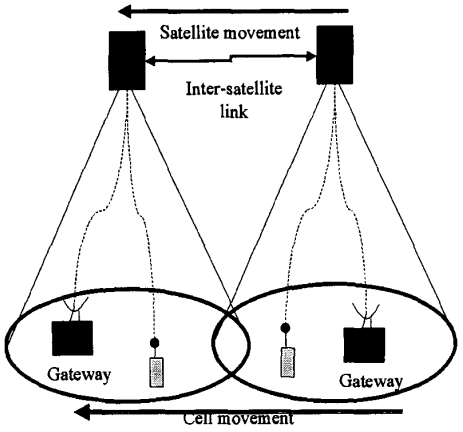


**Figure 2-***Wireless Communications via LEO satellite networks.*

In this paper, we focus on routing challenges in the provisioning of deterministic QoS guarantees such as delay jitter bounds for real-time sustained Constant Bit Rate (CBR) and Variable Bit Rate (VBR) types of user traffic in LEO

systems with satellite-fixed cells. Specifically, we propose a novel routing methodology, which predicts the traffic load on each of the inter-satellite links by exploiting the deterministic LEO satellite topology and location information of users. This information is used to foresee future bottlenecks on possible routes between any pair of end-users. Multiple different paths are determined and are maintained for the duration of the call for avoiding these bottlenecks. If the new route between the end-users required due to user handovers, is longer than the prior established route, the proposed protocol is intelligent enough to reserve a higher bandwidth on the longer route to compensate for longer propagation delays and thereby, satisfy the delay jitter bound. User applications that may require such services include video playback applications with relatively small buffers or voice applications.

## 2. System Model and Problem Definition

### 2.1. Satellite Topology

In the LEO satellite system considered, satellites are moving in '*P*' circular polar orbits. In order to provide global earth coverage, each orbit has *S* satellites. Every satellite has four inter-satellite links, which connect the satellite to its neighboring satellites both in the adjacent orbital plane and in the same orbital plane. It is assumed that the inter-satellite links are reliable and exist for the complete duration of the satellite's orbital revolution. This assumption results in a toroidal mesh architecture for the LEO network as depicted in Figure 3. The satellites have single spot beams, and satellite footprints are non-overlapping and cover a square area. It is also assumed that all satellites move in the same direction with constant speed. Since the earth is round this assumption is only reasonable if a relatively small region is observed. Given the small size of the satellite footprints and the high speed of the satellites (~25000 km/h), it is realistic to assume that ground terminals are stationary in this specific environment. All hand overs are caused by the mobility of the LEO satellites rather than by the motion of the ground terminals. The duration of a connection is typically much less than the orbital period of a satellite (~100 minutes). The rotational displacement of the earth is assumed to be negligible during the lifetime of a connection. However a connection might be handed off from one satellite to another, multiple times during its life span. Users are assumed to be equipped with a location determination system such as the Global Positioning System (GPS).

All satellites in the same orbit cover exactly the same orbital coverage area during a revolution. However, at a given time, each satellite handles traffic from a portion of this orbital coverage region. The user traffic might be non-uniform with respect to both time and location. As the satellite moves along its orbit, the number of users and thus the amount of traffic it serves changes. This change in the amount of user traffic served by a satellite may cause blocking of some of the handover calls due to either the non-availability of the ground user-to-satellite up/down wireless links, or insufficient capacity on ISLs on the route connecting the end users. We assume that there is a Medium Access Control (MAC) layer protocol that ensures the availability of the user-to-satellite links if a path between the end-users is feasible. Thus, we restrict ourselves to the LEO constellation network formed by the inter-satellite links.

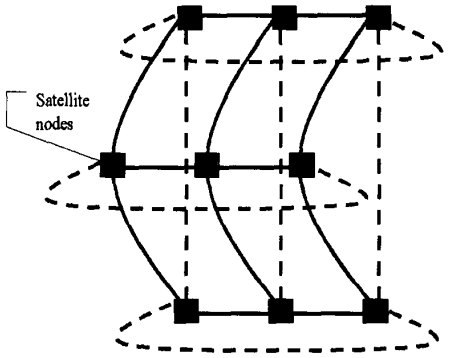


**Figure 3-** *Representation of the LEO satellite topology as a full-mesh topology*

The change in user traffic due to satellite motion is reflected in carried traffic on the inter-satellite links. The traffic on the inter-satellite links change even when the user traffic is static in time. The following properties state that if the user traffic is static the change in carried traffic by each satellite is periodic.

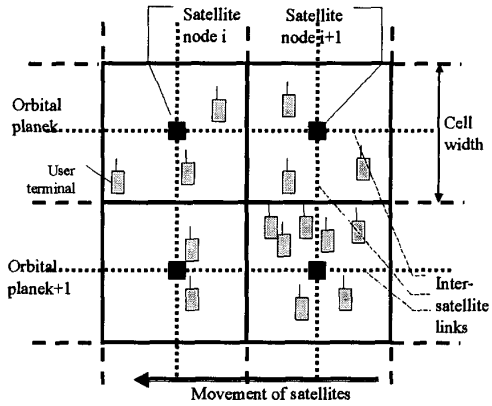


**Figure 4-** *The representation of satellite footprints and non-uniform user traffic.*

**Property 1:** *If the user traffic is static with respect to time, the traffic load on a satellite changes periodically.*

**Proof:** Let $A(t)$ denote the area covered by the footprint of satellite $i$ at time $t$. Assume that the satellite orbits the Earth in $T_0$ time units, and neglect the Earth's rotational motion. At time $t+ T_0$, $A(t+ T_0)=A(t)$, and since the user traffic is static, the property is proven.

Let $T_S$ be the time it takes for a satellite to travel a distance equal to the width of a satellite cell. Then, we can also state the following property.

**Property 2:** *If the user traffic is static, and the satellites footprints are non-overlapping, the user traffic serviced by a satellite 'i' on an arbitrary orbital plane k at time t, is same as the traffic serviced by the next satellite '(i+1)' in the same orbital plane k at time* $(t+ T_S)$.

**Proof:** Let $A_i(t)$ be the area covered by the footprint of satellite $i$ (Figure 4). Then, $A_i(t) = A_{i+1}(t+ T_S)$, since it takes $T_S$ time units for a satellite to travel a distance equal to the width of a satellite cell, and it is assumed that footprints of satellites are square, contiguous and non-overlapping. Since the user traffic is static, the property is evident.

## 2.2. User Traffic

In realistic mobile satellite systems, users are geographically dispersed in a non-uniform fashion. For example, urban areas may be more densely populated with LEO satellite users when compared to rural areas. As the LEO satellite moves along its orbit, it must service as many users that are in its coverage area, as possible. The effects of non-uniform geographical user traffic distributions in LEO satellite networks have not been investigated extensively. As explained in the previous sections, non-uniform user traffic load on the satellites may cause changes in the traffic on inter-satellite links, which may result in unexpected dropping of some of the user calls or packets.

In this paper we consider the delivery of guaranteed services to the users. Guaranteed services require that the packets of a call arrive within a pre-specified guaranteed delivery time and that the packets will not be discarded due to queue overflows. Of course, this service is only possible provided the call's traffic adheres to a specified traffic profile. This service is intended for applications, which need deterministic guarantees that a packet will arrive no later than, or much sooner than a certain time at its destination. The user traffic conforms to a token bucket with bucket size $b$, and token rate $r$ [10]. A user call will request bandwidth $R$ on the links connecting the end-users. The definition of guaranteed service relies on the result that the delay[1] of a call described by the token bucket $(b,r)$, and being served by a link with bandwidth $R$ is bounded by $b/R$ , for $R>r$ [10].

Two kinds of user traffic are of interest: Sustained Constant Bit Rate (CBR), and Variable Bit Rate (VBR). CBR type user applications create packets with equal constant inter-arrival intervals. Applications of this type are not bursty, and the token-bucket at the source can be represented with a token bucket size of one token. VBR type applications; however, are bursty in nature, with the burstiness characterized by the token bucket size.

Delay that a packet endures consists of two parts: a fixed delay (transmission, propagation delays) and a variable queuing delay. In terrestrial wireline networks the fixed delay mainly consists of transmission delays, and propagation delays are negligible. However since distances between the satellites are large, propagation delays in satellite networks are comparable to queuing delays encountered by packets in transit. In order to provide guaranteed service in LEO satellite networks, the fixed delays should be taken into account while providing QoS. Two error terms, $C$, and $D$, represent the way in which the real network implementation of guaranteed service deviates from the fluid model [10]. The error term $C$ is a *rate-dependent* error term, which represents some fixed delays that a packet in a flow might experience. However, this term is dependent on the rate allocated to the flow. Examples of $C$ are the time taken to transmit a packet in a TDMA based system, or the time taken in serializing a large packet, broken up into ATM cells. The error term $D$ represents the worst case *non-rate based* transit time variations such as transmission, and propagation delays.

The bandwidth, $R$, that a connection receives depends on the error terms $C$, $D$, and the end-to-end delay requirement, $D_{req}$ of each packet, and is given by the following equation:

$$R = \frac{b + C_{tot}}{D_{req} - D_{tot}} \qquad (1)$$

where $D_{tot}$ is the total end-to-end propagation delay, and $C_{tot}$ is the total rate dependent delay experienced by a packet belonging to the call [10]. In the considered LEO system, $C$ represents the maximum duration of time that a packet has to wait at the head of the queue of an outgoing inter-satellite link. Note that, $R$, is the amount of bandwidth that is required to guarantee the delivery of packets to the destination within a maximum time of $D_{req}$ time units. Essentially, by increasing $R$ we can reduce the total time that a packet spends in a queue. Thus, if the end-to-end propagation delay between two end-users of a particular call when a particular route is chosen, is large, then in order to satisfy the required delay bound for the call, we have to allocate a larger bandwidth to that call on that route, compared to when the call is routed on a different route with smaller end-to-end propagation delay.

[1] This delay is derived by using a fluid model [10]

The methodology adopted in this paper is a variant of the specifications in [10]. We do not incorporate traffic re-shaping at intermediate nodes. Furthermore, we enforce the rule that the same bandwidth be available on all the links on the route unlike in the *Integrated Services* approach. It is also to be noted that queuing at each node is based on a FIFO scheme and the only means of providing quality of service is by using an admission control policy which blocks new connections if the requested bandwidth is not available on a route. In the subsequent section, we describe the routing problem and due to the fact that a satellite serves different sets of users at different times it is possible that the total allocated bandwidth on a particular link might actually exceed the capacity of the link. In such a case, a percentage of packets might not meet their delay bounds. Thus, the notion of *guaranteed service* is *loose* in some sense.

### 2.3. Routing Problem

As mentioned earlier, current terrestrial routing protocols are not capable of providing QoS guarantees in LEO satellite networks with satellite-fixed cells, due to the inherent time-variance of the user traffic on the inter-satellite links. For example, if we implement a terrestrial routing protocol in LEO satellite networks, then we would find a single route between the entry and egress satellites of the end-users by only considering the loading of the links at call set-up. Although the link capacities may be sufficient to accommodate the call at the call set-up phase on the determined route, this same route may not be able to maintain the requisite QoS for the entire duration of the call since, the loading on the links change in time as the users serviced by each satellite change as the satellites move along their orbits. Hence, a new routing protocol that takes the changes in the loading of the inter-satellite links due to the motion of the satellites into account, is needed.

Our goal is to maximize the total number of calls that are satisfactorily serviced by the network, while maintaining the QoS requirements that are required by each of these calls.

The intuitive and somewhat naïve solution to this problem, is to find a new route for the call, whenever the original fails due to bottlenecks on the links. Although such a solution is feasible in a terrestrial wireless network, where handovers are infrequent and random, it is not the optimal solution for LEO satellite networks. Determining a whole new route as and when such situations are encountered in a LEO network may cause high messaging traffic and processing load on the network, and long interim delays for the ongoing call. Rather than re-routing the call, one could make sure that bottlenecks do not occur or are minimized. This could be made possible by reserving some portion of the bandwidth on the inter-satellite links just for handover calls. The drawback of this solution is that the reserved portion of the channel is underutilized.

These drawbacks can be overcome by exploiting the *predictability* of the LEO satellite topology. The LEO satellite topology at an arbitrary instant can be determined from the information on the connectivity of the constellation, and the satellite velocities. The users in the footprint of each satellite at any particular instant can also be determined by using the location information of the users. Thus, the total traffic that needs to be routed by each satellite at a particular time in the near future can be predicted. This information can be used at call set-up to intelligently determine *multiple routes* for the same call that help avoid *predicted* bottlenecks on the links. The advantage of this method is that the processing delays and the messaging overhead incurred due to handover route recomputation for an ongoing call are avoided. At the same time, only the required bandwidth is reserved at the appropriate times, and thus the bandwidth utilization is improved.

## 3. Predictive Routing Protocol (PRP)

The limited on-board processing capability and the high mobility of the satellites require that the routing function be performed at "ground gateways." The gateways store general network information such as the available bandwidth on each inter-satellite link, and the location and traffic patterns of users. The user traffic information will include for each call, the type of the user application (such as CBR or VBR traffic), the rate at which packets are generated, and the delay or delay jitter requirements for that call.

### 3.1. Criteria for Routing

When a user requests a new connection (to be established with another user), this request is forwarded to the gateway station. The request message reports the locations of the source and destination users, and the requested delay bound.

In determining a route between a pair of end-users in a LEO satellite network, the following should be taken into account:

- The network resources, in particular, the capacity of the inter-satellite links is limited.
- There are an infinite number of calls with different QoS requirements. In this paper, we consider the different QoS requirements as the various delay jitter bounds requested by calls. If a call is accepted to the system, its QoS requirements should be satisfied during the lifetime of the call.
- The network resources should be used efficiently: The messaging and signaling overhead of the routing protocol should be minimized.
- Memory and on-board processing capabilities of the satellites are limited. Thus, the amount of information stored for each connection at any satellite should also be minimized.

One optimization criterion could be the maximization of the ratio of the mean number of calls that can be serviced by the system to the total number of calls requesting service at a given time, such that above requirements are satisfied.

Initially, the gateway determines a route for a given call according to the available satellite link bandwidths observed at the instant that the call request is received (say at $t_0$). Due to satellite mobility, at time $t > t_0$, the satellite may serve users who may be required to use the same links as those used by the call under observation. This may result in an increase in the load (or worse a congestion might result) on these inter-satellite links. Any new call accepted should not degrade the QoS of on-going calls. That is, calls that are in progress should have priority over new calls. In order to ensure that the quality of service of on-going calls is not degraded, the route chosen for a new call should be such that, it does not cause congestion.

In order to predict future loads on the inter-satellite links, the link state information (available bandwidth on the links, the routing table for calls serviced, etc.) which are obtained from satellites, are used by the ground gateways. The residual bandwidths on the links along a determined route are checked to ensure that there is always sufficient bandwidth for the call, for all times, $t$, such that $t_0 < t < t_0 + T_S$. If the route cannot accommodate the call at some time $t_1$,[2] $t_0 < t_1 < t_0 + T_S$ since the minimum bandwidth on the route is less than the required bandwidth, a new route for the call is determined. If this second path is also infeasible at some time $t_2$, $t_0 < t_1 < t_2 < t_0 + T_S$, then another route for the same call is determined. The procedure is repeated until a feasible path for all $t$, $t_0 < t < t_0 + T_S$, is determined. If no feasible path can be found for any period of time between $t_0$ and $t_0 + T_S$ then the call is blocked. Thus, the protocol yields a set of paths $S = \{p_0(t_0), p_1(t_1), \ldots, p_n(t_n)\}$, where $t_0 < t_1 < \ldots < t_n < t_0 + T_S$, and $t_i$ denotes the time at which we start using the path $p_i$ to route packets from source satellite to the destination satellite.

It is sufficient to determine a set of paths for the time duration $[t_0, t_0 + T_S]$, since from property 2, we know that the traffic currently being served by a given satellite say satellite *'i'* will be served by satellite *'i+1'* on the same orbit after $T_S$ time units. In other words, satellite *'i+1' effectively inherits* satellite *'i'*'s coverage after $T_S$ time units.

### 3.2. Routing Protocol Formulation

When a new call request is received by the gateway, the gateway determines the location of the requesting users with respect to a reference framework. The *reference framework* is arbitrary, and represents the locations of the satellites and their respective footprints on the Earth, when the system is initialized. From this framework, the gateway determines the end-satellites that would have been servicing the requesting users at the initialization time. Then, from the known fixed velocities of the satellites, the satellites that are currently servicing the end-users are determined. It is important to note that for the routing purposes, the reference framework can be considered to be identical at times $t = k.T_S$, $k = 0, 1, 2, \ldots$. Thus, note that the gateway can perceive the network as a network of static *virtual nodes*, although the satellite, which represents a particular *virtual node*, will change every $T_S$ time units. However, the load on the satellites and the traffic on the inter-satellite links change for times, $t$, such that $T_S > t > 0$, and the best routing decision at time $t_1$, $T_S > t_1 > t > 0$, may be different from the one at time $t = 0$. The motivation for using a reference framework is to reduce the total number of different states that are considered for routing purposes.

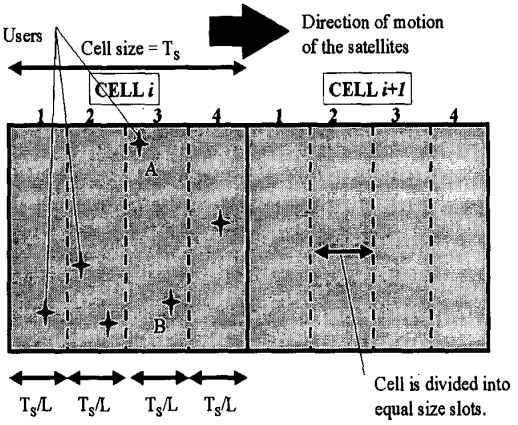


**Figure 5-***The approximation of the geographical coordinates for routing purposes*

The link state information for every satellite changes continuously in time. However, continuously gathering and storing the link state information in such a dynamic environment is prohibitively expensive. For this reason, the satellites gather and forward the link state information periodically with some pre-determined period $T_i$. $T_i$ depends on the required precision in terms of proximity of the approximated discrete-time system to the original continuous time system, and the available storage and messaging facilities in the network. As $T_i$ gets smaller the estimation of the future occurrence of the bottlenecks will get better; however, the processing delays and the storage requirements may increase considerably. The link state information is also periodic with period $T_S$, in the reference frame. In fact, if the link state information is collected with reasonable granularity, it is expected that the total storage requirements per satellite will not be high.

Let each satellite cell be divided into $L$ equally sized slots as depicted in Figure 5. Assume that the satellites gather

[2] Since it is predicted that many calls will use the links in this route at $t_1$.

and store the link state information once every $T_i=T_S/L$ time units. Since the link state information exchanged is limited, the continuous movement of the satellite footprints is viewed as discrete jumps by the routing functions. At each jump, the current satellite cell is *offset* by a slot. Thus, the gateway perceives a *cycle* consisting of a reference cell (corresponding to the reference framework) and *(L-1)* offset cells (Figure 6). At each *offset cell*, the routing function considers only the users that are in the coverage area of this offset cell. Thus, according to this discrete-time model, a satellite traverses its orbit by jumping from a slot to the next in $T_S/L$ time units.

A specific example, for the link state granularity is shown in Figure 5, where $L=4$. For example, assume that a particular satellite has a coverage area consisting of slots 1,2,3,and 4 of cell $i$ (according to reference frame) at time $t=t_0$ (as shown in Figure 5). As the satellite moves along its orbit, its *offset cells* coverage will consist of slots 2,3, and 4 of cell $i$ and slot 1 of cell $i+1$ at time $t=t_0+T_S/L$, slots 3 and 4 from cell $i$ and slots 1 and 2 from cell $i+1$ at time $t=t_0+2T_S/L$, and slot 4 from cell $i$ and 1,2 and 3 from cell $i+1$ at time $t=t_0+3T_S/L$. The process repeats itself for every $T_S$ time units.

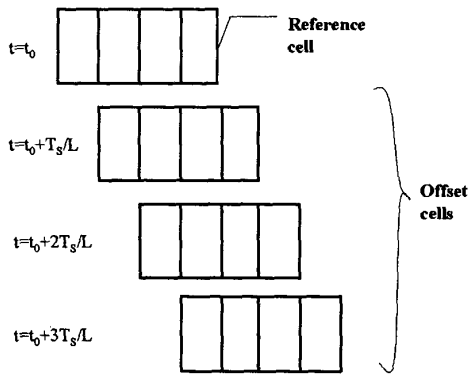


**Figure 6-** *Representation of reference and offset cells.*

Let user A be located in slot 3, of a cell of size $L=4$ slots, at time $t=t_0$, as shown in Figure 5. Users A and B, which are in the same slot according to the reference framework, are considered to be at the same location when the protocol is implemented. However, user B will be handed over to the next satellite at a time later than the time at which user A is handed over. This problem in handling the location of the users with infinite accuracy, may cause some of the routing decisions to be sub-optimal, but we expect the consequences of these situations to be negligible, if $L$ is chosen to be sufficiently large. However, increasing $L$ results in more frequent link state updates, resulting in increased messaging and processing overhead in the network.

In every *offset cell*, the satellite network can be represented by a graph $G=(V,E)$, with a vertex representing a satellite, and an edge representing an inter-satellite link, connecting two adjacent satellites. Every edge has a weight, which is equal to the total residual bandwidth that is available on the corresponding inter-satellite link.

The objective is to accommodate as many calls as possible within the constraints of network capacity. This is equivalent to finding a routing framework that would distribute the total traffic in the network in the most balanced fashion, or the framework that would minimize the load on the most congested link in the network [8]. In order to achieve this goal, packets belonging to a new call are carried on routes *that maximize the minimum residual link capacity* in the network.

The route chosen for a connection does not have to follow the minimum-hop path, as long as the connection's QoS is satisfied. Thus, if it is necessary to change the route due to satellite motion, the bandwidth that is required on the alternate route may be higher than the bandwidth being allocated on the current route, in order to compensate for the difference in propagation delays. In this case, in order to seamlessly transit between the routes we not only need to compute the new route but also need to compute the additional resource required on that route, and allocate it. This may require switching of other connections to longer routes, if necessary, and if their delay constraints are not stringent. The required bandwidth, R, for a particular connection with a required end-to-end delay bound, Dreq, is given by equation (1).

In order to minimize both the control traffic messaging frequency and the routing table storage requirements in the satellites, we also need to minimize the number of route changes for a particular connection required as the satellite footprints move. The paths that maximizes the minimum residual bandwidth for each offset cell $l=1,2...,L$, may be quite different from one another. Since the objective is to minimize the amount of information that needs to be stored/propagated for a single flow, the protocol should ensure that the computed set of routes consists of a single route or routes which do not differ from each other by much. In other words the number of link changes required to transit from one route to another within the set should be minimized. Note that, it is sufficient to store the original route, the changes with respect to the original route and the times at which these changes will take effect. These entities correctly define the complete set of paths determined by the protocol. In order to compute this optimal set of routes, for each offset cell we determine $k$ ordered paths that maximize the minimum residual bandwidth for that offset cell $l$, $0<l<L$. Let us represent these $k$ ordered paths by a set $\{k_l\}$. We then pick one path from each set $\{k_l\}$ such that the *combined set* of paths we pick is the best in the sense that the number of link changes required as we transit among the paths in our chosen combined set is minimum as compared to the number of link changes required when transiting among the paths in any other

combined set which contains one path from each $\{k_l\}$. This selection of the optimal combined set of paths is achieved through the use of a cost function that weighs the importance of the resemblance of the different paths chosen for different offset cells, to the maximization of the minimum residual bandwidth in the network.

Let, $c_i(l)$ be the current link capacity on link $i$ at offset cell $l$, $d(p)$ be the end-to-end propagation delay for a chosen path $p$, $R_p$ be the bandwidth required for the new connection on path $p$, and $r_p(l)$ be the residual link capacity on path $p$ if the connection is admitted in the offset cell $l$. Then, the Predictive Routing Protocol (PRP) can be as follows:

*(a) Convert source and destination user location information (which is in longitude and latitude) to the reference cell and slot information, i.e. identify the reference cell and the slot within the cell where the user is located.*

*(b) For $l=1,\ldots, L$ { for all offset cells }*
- *Determine the reference end satellites serving the users.*
- *Determine all paths with J or less number of hops connecting these end satellites.*
- *For each path,* p, *calculate the minimum residual link capacity, which is* $r_p(l) = c_i(l) - R_p$, *and where,* $R_p=(b+C_{tot})/(D_{req}- d(p))$
- *Determine* k-ordered paths for offset cell 1 , $\{k_l\}$ that *maximize the residual link capacity.*

*(c) For each offset cell $l=1,\ldots,L$, pick a path among* k-ordered paths, $\{k_l\}$ *determined above. Call this combined set of picked paths $S_m$, where $m=1,\ldots,k^L$.*

*(d) Determine total number of link changes required on this set of paths, as we transit among the paths and denote it by $H_{Sm}$. Repeat from (c) for each different combinations of these* k-ordered paths.

*(e) Determine the overall reward for each combination set $S_m=\{p_1(i_1), p_2(i_2),\ldots, p_L(i_L)\}$, where $i_l=1,\ldots, k$ is the path chosen for offset cell l, by:*

$$Reward(S_m)=\sum_{l=1}^{L} r_p(l) - W.\ H_{Sm}$$

*where W is a constant used to weigh the relative importance of having few link changes on the route for the call as we transit offset cells, with respect to the balancing of the user traffic.*

*(f) Choose the $S_m$ that maximizes the reward.*

After a path for every offset cell l=1, ..., L is determined for the connection, the necessary bandwidth for each path to satisfy the QoS requirements, is reserved along the links forming the paths. However, since the reservation is made only for a slot duration, during other offset cells, the same bandwidth can be possibly used for other connections. Thus, the network utilization is improved.

When a packet is generated, the offset cell in which the packet is generated is determined. Then, the route corresponding to this offset cell, is used to forward the packet from the virtual node serving the source user terminal to the virtual node serving the destination user terminal.

### 3.3. Packet Forwarding

The user packet headers have the following information fields to be used by the protocol:

1. Source user address: The address of the source user is specified according to the cell and the slot the user is located at with respect to the reference framework.
2. Destination user address.
3. Flow identification number: Since the addressing scheme does not refer to an exact location, but rather a region, there may be multiple flows with the same source and destination user address. In order to differentiate between the flows, each flow is identified with a different unique flow identification number.
4. Timestamp: The packets are time stamped[3] by the source node. This information is used by the intermediate nodes to identify the offset cell the packet is generated in.

| Source user address: cell location and slot number | Destination user address: cell location and slot number | Flow id | Timestamp: Time the packet was created |
|---|---|---|---|

**Packet header for PRP protocol**

**Figure 7-** *PRP packet header*

When a packet is received at an intermediate node, the node examines the source and the destination addresses and the flow identification number, and determines the set of paths used by the connection. From the timestamp, the intermediate node determines the offset cell that the packet is generated in. The route corresponding to *this offset cell* is used for the transfer of the packet from the source user terminal to the destination user terminal.

The packet-forwarding protocol formulated as follows:

*(a) From the time stamp on the packet, determine the offset cell in which the packet is created.*

[3] The time-stamp indicates the time of creation of the packet

*(b) The path corresponding to that offset cell is looked up in order to determine the next hop.*

## 4. Simulation Results

For simulations, UCB/LBNL/VINT *Network Simulator (ns2.1v6)* has been used [11]. In order to test our LEO routing algorithm, we extended the simulator by including modules implementing LEO satellite handoffs, and our own routing protocol.

The LEO network that is considered in the simulations consists of 25 satellites. There are 5 orbital planes each with 5 satellites. The resulting network is a 5x5 toroidal mesh network. The cells are adjacent and square, and the time taken by a satellite to traverse one cell is 1.0 second. Every cell is divided into 5 equal length slots. Thus, the satellite traverses a slot in 0.2 seconds. In the following simulations we consider the weighting factor, $W$, to be zero.

A call is blocked when no combined set of paths (as defined earlier), can be found for a particular call. There are two performance metrics considered in the simulations: New call blocking probability, and the packet dropping probability. A packet is dropped when it violates the call's delay requirement. As mentioned earlier, due to the fact that traffic is not reshaped at intermediate nodes, and because the satellite serves different sets of users at different times, it is possible that total bandwidth of the connections being routed on a particular link actually exceeds the capacity of the link. In such a case, packets would miss their deadlines and hence be dropped.

In the first part of the simulation, connections are voice or video calls that are Constant Bit Rate (CBR) flows. The end-users are distributed uniformly in coverage area. Calls arrive according to a uniform distribution with a mean inter-arrival time of 5.0 seconds. The duration of each call is the outcome of a uniform distribution with a mean of 2.0 seconds. Each call generates 100 packets per second. The size of a packet is 210 bytes. Every call has the same delay requirement $D_{req}$; that is every packet of this call should be delivered at the destination in $D_{req}$ seconds. Average propagation delay on each inter-satellite link is 20 milliseconds.

Figure 8 depicts the total number of calls with a given delay requirement, that can be serviced by the LEO satellite network with PRP. The results are parameterized according to the acceptable blocking probability. The average packet dropping probability for any given blocking rate is approximately 2%. The bandwidth of an inter-satellite link is 1Mbps. Clearly, as we relax the delay requirement, the total number of calls that can be serviced increases. If the blocking rate requirement is quite strict, the number of users that can be served by the system is quite low. For example, for $D_{req}$=80 ms, no flows can be admitted if the blocking rate requirement is as low as 5%.

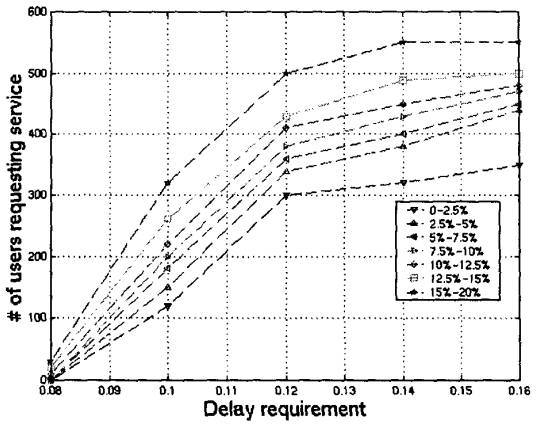


**Figure 8-** *Total number of users that can be serviced by the system vs. required delay. The graph is parameterized with respect to the blocking probability that can be accepted. CBR user traffic distributed uniformly among the total coverage area.*

From Figure 8, we can see that the maximum total number of users that can be serviced by the LEO satellite network with the PRP routing algorithm is around 450 if a blocking probability of 15-20% is permissible. Figure 8 can be used to implement an admission control policy to restrict the number of flows for achieving a desired blocking probability and delay requirement when PRP is used.

In Figure 9, we compare the performance of the PRP routing algorithm with a non-predictive routing protocol (NRP), and IP routing algorithms. IP routing algorithm routes packets according to the Dijkstra's shortest path algorithm. In this context, the computed route is the same as the minimum hop path between the end-satellites. The shortest path between the satellite nodes are calculated at the beginning of the simulation. The IP routing does not consider the loading on the links or reserve any bandwidth for the call. It may hence route packets along congested links.

When a new call arrives, the non-predictive routing protocol (NRP) determines a path between the end-satellites serving the corresponding terrestrial end-users, by considering the traffic on the inter-satellite links only at the call set-up. The path determined by NRP maximizes the minimum residual link capacity on the LEO inter-satellite links at call set-up. The bandwidth that is needed to satisfy the delay requirement of the call is also reserved at call set-up.

In Figure 9, the total number of users requesting service is 200. The delay requirement for all calls is 100 ms. In this simulation, blocking and packet dropping probabilities are determined by varying inter-satellite link capacities. The PRP protocol has a higher blocking probability than the NRP algorithm, since the PRP algorithm provides more stringent QoS guarantees than the NRP algorithm. We re-emphasize that unlike the NRP, the PRP takes into account the dynamics of traffic changes due to satellite mobility while computing the optimal routes. This situation is apparent from the packet dropping rates shown in the figure. The NRP algorithm has much a higher packet dropping rate, when compared to the PRP algorithm. The IP routing algorithm is not expected to provide guaranteed service, and hence, many of the packets may be expected to violate their delay requirements as may be seen from the figure.

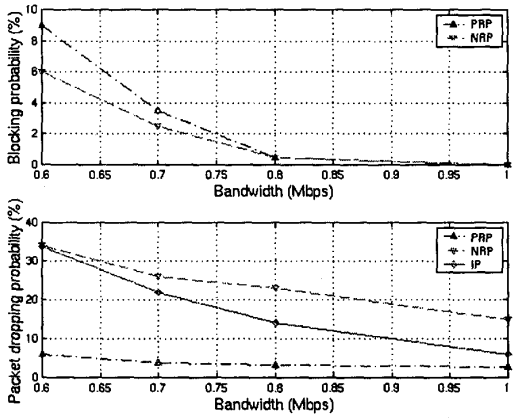


**Figure 9-** *Comparison of PRP with respect to the NRP and IP routing algorithms. Total number of users requesting service is 200.*

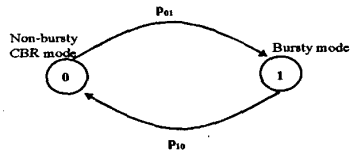


**Figure 10-** *The model for VBR source. $p_{01}$=0.2, $p_{10}$=0.2. The token bucket size is 10.*

In Figure 11, we present the simulation results for VBR type calls. In this example, the total number of users requesting service is 200, and the deterministic delay requirement for the packets is 100 ms. The VBR source is modeled as a two-state Markov Chain as depicted in Figure 10. When the Markov chain is in state 0, it generates packets with equal constant inter-arrival times of 10 ms. The source switches to state 1 with probability 0.2. When the Markov chain is in state 1, it generates packets with inter-arrival times conforming to an exponential random variable with mean 3.33 ms. The token bucket size *'b'* is 10. The token rate *'r'* is 8 ms. The VBR source switches back to state 0 a with probability of 0.2.

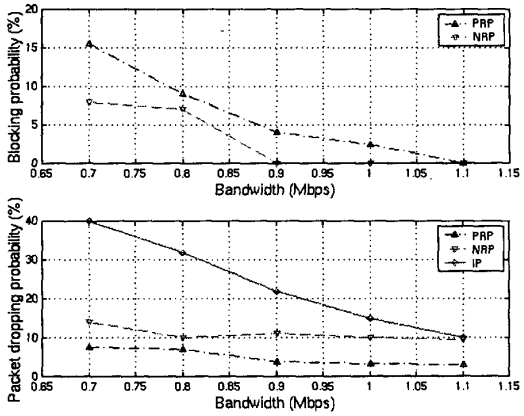


**Figure 11-** *Comparison of PRP with NRP and IP routing algorithms for a VBR source. VBR source has a bucket size of 10. Total number of users is 200.*

The results in Figure 11, show that, as expected the PRP algorithm has a higher blocking probability when compared with the NRP algorithm, but the packet dropping probability of PRP is much lower than the NRP.

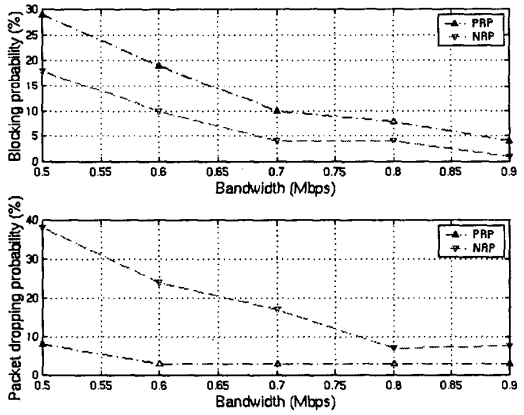


**Figure 12-** *Comparison of PRP with NRP when the users are distributed non-uniformly in the coverage area. Total number of users is 200.*

In Figure 12, we observe the performance of PRP as compared with NRP for a non-uniform user traffic distribution. The abscissa of the cells that the source and destination terrestrial users reside at, is determined according

to the following probability vector $V$=(0.066, 0.13, 0.2, 0.26, 0.33), where $i^{th}$ entry of $V$ refers to the probability of having a particular user in one of the five cells corresponding to the $i^{th}$ column of the 5x5 reference mesh network. The ordinate of the cells that determine as to where the source and destination terrestrial users reside, follow a uniform distribution. The connections are CBR flows, with a packet generation rate of 100 packets/second. The deterministic delay requirement for all the calls is the same, which is 100 ms. The propagation delay of each inter-satellite link is 20 ms. From Figure 12, we observe that, as expected, PRP outperforms the non-predictive routing algorithm in terms of the packet dropping rates.

## 5. Conclusions and Future Work

Broadband LEO satellite networks will complement current terrestrial broadband networks, and will provide service to users regardless of their location. In this paper, we have presented issues related to routing in a broadband LEO satellite network with an emphasis on delivering deterministic QoS to users. The traffic on the inter-satellite links between the satellite nodes change dynamically as the satellites move along their orbits. In order to deliver QoS guarantees, these variations in traffic should be taken into account, while designing a routing algorithm.

We have proposed a predictive routing algorithm that exploits the deterministic nature of the LEO satellite topology in order to deliver QoS guarantees. It has been observed from a set of detailed simulation examples that the PRP can provide strict QoS guarantees without overreserving capacity on the inter-satellite links. An admission control curve has been obtained which may be used to ensure that the desired QoS metrics may be guaranteed.

The implementation of the algorithm is currently limited to point-to-point unicast connections. Future LEO satellite networks should be able to support efficient multicast connections as well. Supporting multicast connections with QoS guarantees is an interesting research problem requiring further research. Our findings can also be extended for hybrid LEO-GEO, and LEO-wireless ad-hoc communication networks. To investigate issues in QoS routing in such hybrid environments are among our future research goals.

## 6. REFERENCES

[1] A. Jamalipour, *Low Earth Orbital Satellites for Personal Communication Networks*, Artech House, Inc, 1998.

[2] M.A. Sturza. "Architecture of the TELEDESIC Satellite System," *Proceedings of International Mobile Satellite Conference*, p.p. 212-218, 1995.

[3] F. Dosiere, T. Zein, G. Maral, and J.P. Boutes. "A Model for the Handover Traffic in Low-Earth Orbiting (LEO) Satellite Networks for Personal Communications," *International Journal of Satellite Communications*, 11:145-149, 1993.

[4] E. Del Re, R. Fantacci, and G. Giambene. "Handover Requests Queuing in Low Earth Orbit Mobile Satellite Systems," *Proceedings of the 2nd European Workshop on Mobile/Personal Satcoms*, p.p. 213-232, 1996.

[5] M. Werner, C. Delucchi, H.-J. Vogel, G. Maral, and J.-J. De Ridder. "ATM-Based Routing in LEO/MEO Satellite Networks with Intersatellite Links," *IEEE Journal on Selected Areas in Communications*, 15(1):69-82, Jan. 1997.

[6] B. A. Akyol and D. C. Cox. "Rerouting for Handoff in a Wireless ATM Networks," *IEEE Personal Communications*, 3(5):26-33, Oct. 1996.

[7] K.Y. Eng et al. "A Wireless Broadband Ad-Hoc ATM Local Area Network," *Wireless Networks*, 1(2):161-174, 1995.

[8] J. F. P. Labourdette and A. S. Acampora, "Logically Rearrangable Multihop Lightwave Networks," *IEEE Transactions Communications*, vol.39, no. 8, pp.1223-1230, Aug.1991.

[9] H. Uzunalioglu, W. Yen, and I. F. Akyildiz, "A Connection Handover Protocol for LEO Satellite ATM Networks". *Proc. of ACM Mobicom '97*, Budapest, Hungary, Sept. 97.

[10] S. Shenker, C. Partridge and R.Guerin, "Specification of Guaranteed Quality of Service," *RFC 2212.*

[11] VINT Network Simulator *(ns)*, version 2.0. University of California, Berkeley, Lawrence Berkeley National Laboratories, 1998, http:://mash.cs.berkeley.edu/ns/ns.html