

UNIVERSITÀ DEGLI STUDI DI PISA
DIPARTIMENTO DI INFORMATICA
DOTTORATO DI RICERCA IN INFORMATICA

PH.D. THESIS

Integrating LEO Satellite Constellations into Internet Backbone

Yan He

SUPERVISOR
Susanna Pelagatti

May 7, 2007

Addr: Largo B. Pontecorvo 3, I-56127 Pisa – Italy
Tel: +39-050-2213162 — Fax: +39-050-2212726 — E-mail: he@di.unipi.it

for wife Jinghua

Acknowledgments

I want to thank many people who help me to finish this thesis. First of all, of course many thanks to my family. My parents, Jiliang He and Hong Ge, they bring me to this world and let me know how beautiful the world is. My father always gives me useful suggestion for English. My mum, a traditional chinese mother, always worries that I can not take care of myself. Thanks my lovely wife Jinghua, for her love and patience for my complain. I can not survive without her delicious cakes. I love her forever. Thanks to Jinghua's family, my father-in-law Zhenming Gao and mother-in-law Shuyu Li. They give me their daughter, the greatest gift of my life. Also thank Mibao, my cat, he is so funny.

Many thanks to my supervisor, Prof. Susanna Pelagatti. She brings me into this research area. She always supports me and pushed me to go ahead. She gives me lots of brilliant ideas for my work. I can not finish this thesis without her help.

Thanks to people in the Wireless and Mobile Group in Dipartimento di Informatica. Prof. Maurizio Bonuccelli helps me and gives me lots of knowledge at the beginning of my research. Thanks to Researcher Stefano Chessa who helps me a lot for my paper. Also thanks to Francesco Nidito for his help in simulation and to Francesca Martelli for her help. Thanks Francesca Lonetti for her interests.

I would thank the committee of my thesis, Prof. Linda Pagli and Prof. Fabio Panzieri. They give me lots of useful ideas and suggestions.

Thanks to Prof. Amiya Nayak and Prof. Mohan Kumar, the reviewers of my thesis. They spent a great deal of time to read and understand my thesis. They give me very important suggestions to revise my thesis.

Thanks to Prof. Andrea Maggiolo Schettini, the ex-director of our Ph.D school, and Prof. Pierpaolo Degano, our new director. They help me solve lots of problems in my Ph.D career.

Thanks to my football group, Andrea, Guglielmo, Daniele, Giuseppe, and Maurice, for the fun and the improvement of my skill.

Thanks to the lunch group, Sara, Alessandro, Flavio, Razvan, Alessio and Alina for those interesting talks. I wish they always have good appetite.

Finally, I want to thank all those people I did not mention above. Thanks for them help in my life.

Contents

Acknowledgments	3
1 Introduction	11
1.1 Motivation of this thesis	11
1.2 Technical challenges	12
1.3 Satellite routing in the literature	14
1.4 Contribution of this thesis	14
1.5 Outline of the thesis	15
I Background	17
2 Background	19
2.1 Background: New challenge for Ground Internet	19
2.1.1 Present Internet Routing Problems	19
2.1.2 QoS Guarantees for Multimedia Services	21
2.1.3 IP Multicast	22
2.1.4 Mobility Support of Internet Access	23
2.2 Background: Satellites Constellations	23
2.3 Motivation of using LEO constellations	27
2.4 Technical challenges of integrating Internet with LEO constellations .	28
2.4.1 Multiple Access Control	28
2.4.2 Routing	30
2.5 Summary	32
3 Satellite Links and Protocols	33
3.1 Constellations and Intersatellite links	33
3.1.1 Intersatellite links	33
3.1.2 Constellation categories with respect to ISL	35
3.1.3 Onboard switching	37
3.2 Satellite Protocol Stack Overview	38
3.3 MAC Layer	40
3.3.1 Optical vs. Radio Frequency	40

3.3.2	Multiplexing techniques	41
3.3.3	Multiple access techniques	42
3.3.4	Frequency Division Multiple Access	43
3.3.5	Time Division Multiple Access	43
3.3.6	Code Division Multiple Access	44
3.3.7	Demand Assignment Multiple Access	45
3.3.8	Random Access Protocols	45
3.4	Transport layer in LEO constellation	47
3.4.1	Implementing plain TCP	47
3.4.2	TCP Modifications	47
3.4.3	TCP proxies	48
3.5	Summary	49

II Routing in LEO constellation 51

4	Networking in LEO constellation	53
4.1	IP over LEO constellation	53
4.2	Routing Concept and Topology Models	54
4.3	Routing Classifications	55
4.4	Related Work	57
4.4.1	DT-DVTR and other offline routing	58
4.4.2	Predictive Routing	58
4.4.3	Dogleg, Parallel highways and Polar hop routing	59
4.4.4	Greedy routing using buffer information	60
4.4.5	Minimum propagation delay routing	60
4.4.6	End-to-End Delay routing	63
4.4.7	Hierarchical group routing and cluster routing	63
4.4.8	Routing using Markov Decision Process	65
4.4.9	Traffic class routing	65
4.5	Routing algorithms summary	67
5	Our Control Route Transmission (CRT) Protocol	69
5.1	Motivations	69
5.2	Constellation model and traffic model	71
5.2.1	Constellation Architecture	71
5.2.2	Our Constellation Model	71
5.3	Control Route Transmission (CRT) Protocol	73
5.3.1	Stream format	74
5.3.2	Computing the best route	74
5.3.3	Routing a new burst	76
5.3.4	A simple example	76
5.4	Performance evaluation	77

5.4.1	Simulation tools	78
5.4.2	Simulation setting in GeNeSi	79
5.4.3	Simulation Results	83
5.5	Summary	92
6	CRT with priority and bandwidth request (BCRT)	93
6.1	Motivation	93
6.2	CRT with bandwidth allocation (BCRT)	94
6.2.1	Reservation protocol	95
6.2.2	An example with BCRT	97
6.3	Simulation Setting	99
6.4	Simulation Results	100
6.4.1	Experiments with uniform traffic distribution	100
6.4.2	Experiments with non-uniform traffic distribution	103
6.4.3	Time cost for routes computation	107
6.5	Summary	108
7	Conclusion	109
	Bibliography	111

Abstract

Low Earth Orbit (LEO) satellite constellations have been used for ubiquitous and flexible Internet access services. However, a number of problems related to the integration of terrestrial with satellite hosts should be resolved for the effective exploitation of LEO constellations. LEO constellations are different from terrestrial Internet because of its special properties, which result in a lot of problems. A key issue is how to route Internet packets to the LEO constellation. In the thesis (1) the background of LEO constellations was introduced; (2) the obstacles of routing between the satellites and Internet were outlined; (3) The particular problem, which must be solved, is the routing burst stream traffic in LEO satellite constellations. Two novel routing algorithms Control Route Transmission (CRT) and CRT with bandwidth allocation (BCRT) were utilized to address the bursts routing problem. CRT is an adaptive protocol which is able to minimize the congestion in the constellations. BCRT is a CRT extension which is allowed to class the traffic (e.g. video) with different QoS requirements and guarantees. Both of CRT and BCRT work in time epochs. Routes are computed on the basis of a directed weighted graph representing the global traffic traveling in the constellations. Both CRT and BCRT were evaluated via simulation and compared with other proposals in the literatures. The results showed that CRT is a simple algorithm, but the strategy produced by CRT could avoid the congestion and enhance the global resource usage in different traffic conditions. Moreover, the explicit reservation and reroute of BCRT greatly improve the performance of CRT. In particular, the dropping rate of BCRT is very low and the average delivery time is comparable with other proposals in the literatures.

Chapter 1

Introduction

1.1 Motivation of this thesis

Ground Internet new challenges Internet has enjoyed an explosive growth in the past few years. Although the architecture of Internet has been improved so that it allows a better world wide access, recently the proliferation of new applications and the expansion in the number of hosts and users impose new technical challenges to Internet development[Hal97, LMJ98, Li03]. Routing problems, which are due to the inherent vice of the Internet terrestrial infrastructure like routing instability, slow convergency time and limited scalability, become more and more severe. New Internet infrastructure and technologies that can provide high-speed and high-quality services are required[Hal97][Li03]. Moreover, the applications (like YouTube video) that accommodate multimedia and real time services with diverse Quality of Service (QoS) requirements are more and more popular. Such applications need soft to hard guarantees on the actual performance experienced by packets (e.g, delay, jitter, packet loss). It has been recognized that terrestrial Internet structure is particularly unsuitable to accomplish such requirements. Hence many proposals, which modify Internet structure to accommodate multiple traffic flows with diverse QoS, have been made in the literatures [BCS94, BBC⁺98]. However, most proposals would need quite a number of radical changes on Internet structure. Another problem is related to UDP based on IP multicast, which is still open and uncontrolled. UDP's delivery is unreliable and there is no guarantee for the transmission rate, while world wide IP multicast is still an open research issue although there are lots of proposals about IP multicast in a single domain[WD88, Moy94, Bal97, EHT⁺98]. Moreover, the requirement for ubiquitous Internet access has grown enormously. Users expect to be able to access Internet at any time and place around the world[HL01], however, in case of disasters and wars, it is extremely difficult to guarantee global Internet access. For instance, the submarine fiber cables between China and North America were broken because of the earthquake happened in December 2006 so that Chinese were not able to access oversea Internet for two weeks. On the other hand,

outer space scientific research, like Mars explorations, need to access the remote data and the remote pilot equipment in delicate experiments[AAC⁺03]. For reason given above, the appropriate support is required in areas where the deployment of terrestrial infrastructure is very difficult or impossible.

LEO satellites Recently, some authors have suggested that the satellite networks should be used to overcome part of Internet problems[NCD98, HL01, Woo01a]. In particular, Low-Earth Orbit (LEO) satellite networks seem to be especially promising to support ubiquity and mobility as they can be effectively deployed on distant planets, offer global earth coverage and require relatively small antennas which are of the same size and weight as devices used in cellular phone system[Gav97]. Moreover, they possess the inherent broadcast capability due to the small footprint and frequency reusability, and bandwidth-on-demand flexibility based on the MAC layer protocols (FDMA,CDMA) [Gav97, AJ97]. However, the inter-operation between satellite networks and terrestrial Internet infrastructure introduces new challenges and it is far from being completely understood.

1.2 Technical challenges

Integrating Internet with LEO constellations A number of proposals have been reported in the literatures, in which the potentiality integrating LEO satellites with terrestrial Internet was exploited[NCD98, HL01, EAB01, Woo01b]. Two architectures for integration can be considered to be useful.

- a.* considering each satellite as an Internet node and adapting the existing terrestrial protocols to satellite characteristics.
- b.* considering the satellites as a completely different network which has its own protocols and structure.

Most of the proposals in the literatures have followed the approach *a* which result in a body of researches and development about the data communication on the satellite networks in both of ATM and IP implementations [AJ97, HMK98, WDV⁺97, Wer97, Rag02, HS98, NCD98]. Also the majority of this researches concentrates on the pure IP/ATM routing problems when the satellites are involved in providing Internet access service for terminal users. The approach seems not to be very promising as it only can provide the solutions to support ubiquitous Internet access but can not subtract traffic from the Internet backbone.

On the other hand, some recent proposals have followed the approach *b* in which the LEO constellations are considered as an Autonomous System (AS) with its own routing policies, and the work is concentrated on integrating this special AS as an extended cascade structure with the Internet. However, the actual integration

of satellites networks with Internet in this framework is still a largely unexplored area[HL01, Woo01b].

We intend to investigate the actual feasibility of approach *b* for the integration of satellite with terrestrial Internet. The solution of different technical problems in several research areas is required. According to the physical layer point of view, the radio frequency is usually used as communication medium for LEO satellites. The link error rate of unstable wireless connection is higher than that of terrestrial cables. In MAC layer, corresponding to wireless connection, FDMA, TDMA and CDMA are candidates for MAC layer technique in LEO constellation[AJ97, Mar05]. In transport layer, two main problems of LEO satellites are the longer round trip time and higher error rate as compared terrestrial links, which the plain TCP is not able to overcome. In network layer, the high mobility of LEO satellites results in the difficulties to apply IP over LEO constellations. The long delay and unstable constellation topology cause the more severe routing problems, for example, routing instability and slow convergency time. Internet routing problems will be discussed in Section 2.1. On the other hand, only simple routing algorithm can be used for the satellites due to the limited resource onboard of the satellites. The common solution methods are to use IP tunneling and NAT for translating addresses between satellites and terrestrial Internet, and to implement a simple private routing algorithm inside the constellations.

Video support in LEO constellations Recent years, the different solutions of supporting video services have been proposed for satellite and wireless environments. Most of these proposals work on MAC layers. Koraitim and his colleague proposed DMBS, a resource allocation and admission control policy for satellite networks [KT99]. DMBS makes the decision of allocation at the beginning of each control period which depends on the network load conditions. It also monitors the traffic request queues and uses a threshold of regulating the admission control. So and Cho proposed an access control protocol based on residual capacity prediction [SC02]. Cheng and his colleagues analyzed the behavior of CBR using prediction protocol in wireless networks [CM03]. Furthermore, the dynamic reservation with different priorities is introduced into wireless networks based on the different service requirements. Each service has its priority for resource contention. The priority can be set, which depends on the different traffic classifications, for example, video, voice and normal data [FLC01]. In particular, even in the same video traffic class, multipriority is proposed for video partitions to enhance the multimedia capacity[RNJ03]. However, all these proposals focus mainly on the utilization of a single link. In order to optimize efficiency, a global view of all ISLs is required. In this case, the utilization of all ISLs, including ISLs above suburban areas, like deserts and oceans, is helpful to balance the traffic and improve system throughput. On the other hand, QoS guarantee for individual requests is another important issue to be concerned. Therefore, we need to switch to network layer for finding an adaptive routing solution

of these problems.

In present thesis, we focus on routing with NAT and analyze the details of routing problems inside LEO constellations, which concentrate on special issues like routing for message bursts and video.

1.3 Satellite routing in the literature

Different solutions have been proposed to address the routing problem in LEO satellite constellations. These proposals focus on best effort delivery through LEO satellite constellations optimizing different parameters. Offline algorithms [Wer97, HL01, RE00] compute routes with offline information pre-stored in the storage. It is very simple but not adaptive. It is easy for Offline routing to cause congestions. Centralized algorithms like [EKDT00] need a control unit to compute routes, which have the disadvantages of extra communication round trip time to get routing information and it is expensive for routes recovery after failures. Geographic algorithms like Dogleg, Parallel highways and Polar hop routing [KE03, MR97] are very simple but they are not adaptive and easy lead to congestion. The propagation delay time is a key objective function to compute routes. The delay time is utilized to compute routes with two different solutions. One is to compute routes base on the minimum propagation delay between source and destination satellites [HK00, EBA01]. The other one is also based on the propagation delay, but the delay bound requirement is used to provide end-to-end delay guarantee [HYK04, HYK05]. Beside only using LEO satellites, people also try to use MEO satellites together with LEO in hierarchical group routing [CE05, YZL05, LMYW04].

Details of proposed algorithms in the literature will be discussed in Chapter 4.

1.4 Contribution of this thesis

In this thesis, we propose two routing algorithms, CRT and BCRT, for burst traffic on satellite constellations. CRT works in time epochs, in which the periodically exchanged congestion control messages are utilized to build a congestion matrix, then the shortest path is computed on this matrix and the best route for all pairs sources/destinations is obtained. The periodically exchanged information can adjusts routes to avoid congestions in hot spot areas. CRT balances the traffic load over large areas of the constellations. The performance of CRT will be evaluated by simulation using different traffic scenarios. CRT will be compared with other algorithms proposed in the literature. The results will be presented in different emphases like packet lossy percent and delivery time. In particular, the behavior of the overhead due to periodically exchanged control messages will be evaluated.

BCRT is an extension of CRT. It takes care of a bandwidth reservation for QoS traffic on the satellites. Bandwidth is reserved using a distributed protocol before a

burst is admitted in the constellations. Once granted, a bandwidth is guaranteed for the whole transmission. BCRT also works in time epochs. It periodically exchanges information on the status of bandwidth reservation. The information is used to drive the selection of the best route during the bandwidth reservation phase. The performance of BCRT is also evaluated with the simulation.

1.5 Outline of the thesis

The rest of this thesis is organized as follows: In Chapter 2 the existing problems of terrestrial Internet will be analyzed and the general background on satellite constellations will be introduced. Then, in Chapter 3, the satellite protocol stack will be discussed from lower physical layer up to transport layer. Different models of constellations also will be introduced. In Chapter 4 the details of network layer of the constellations will be examined, which focus on the routing problem. Some new related routing algorithms proposed in the literature will be analyzed. In Chapter 5 the system model for our research is presented; then the CRT routing protocol with a simple example is described; finally the simulation setting and results analysis are discussed. In Chapter 6 BCRT, the extension of CRT taking care of QoS and bandwidth reservation, is presented. Also the performance of the algorithm presenting some simulation results is discussed. In Chapter 7 the conclusion and some possible development of the thesis is represented.

Part I

Background

Chapter 2

Background

In this chapter, first of all, the present Internet challenges caused by the increasing access demand of users and high technical requirements will be outlined briefly. Secondly, the different types of satellite constellations will be introduced. Finally, the motivation of utilizing LEO satellite will be discussed, because LEO satellite is selected as the candidate to solve the present Internet challenges. It is important for us to solve the technical problems.

2.1 Background: New challenge for Ground Internet

The constraints due to the growth and complexity of terrestrial Internet infrastructure cause lots of problems for application of new technique. The solution of the problems, for example, Routing, QoS guarantee for multimedia services and Support for IP multicast, requires some novel technique. So the problems of terrestrial Internet will be described particularly, and some solutions proposed in the literatures will be reviewed.

2.1.1 Present Internet Routing Problems

From routing viewpoint, Internet is composed of a large number of interconnected Autonomous Systems (ASs). Each AS constitutes a distinct routing domain which normally also belongs to a single administration domain. Within an AS, the routers communicate each other using one or multiple intra-domain routing protocols such as Distance Vector Routing and Link-State Routing. ASs are connected via the routers which exchange information through inter-domain routing protocols. The most popular inter-domain routing protocol currently used is Border Gateway Protocol (BGP)[Hal97][Li03]. Even though Internet routing issues have been studied for a long time, there are still lingering problems due to the explosive growth of Internet.

Routing Instability The problem of routing instability consists in the rapid fluctuation of network reachability and topology information[LMJ98]. Many reasons may lead to routing instability. When a router detects a link failure or some unreachable network prefixes, it advertises to its neighboring routers that the route is updated. The neighboring routers will propagate the data across the network using the updated information. Since each router makes its own decision with its local policies, one router may receive different updates results for the same peers. On different routers, it may lead to different views about the topology change. Finally, there may be many (possibly divergent) routing updates going on simultaneously which can cause a larger scale of routing instability. The other drawback of routing instability is the implementation and performance of participating routers on Internet. A large number of frequent updates increases cache miss in routers. The above facts in turn increase the load on CPU and the switching latency, as well as the number of packets lost. Similarly, low-end routers may not perform well under heavy update load. When it happens, the performance of packet routing and processing is degraded. The oscillation between alive routers status and dead routers status may be another reason of instability.

Two notable proposals have been used to solve this problem: *Route Flap Damping* (RFD)[VCG98] and *Classless Inter-Domain Routing* (CIDR)[FLYV93].

The major goal of RFD is to reduce router's processing load caused by instability and deter the propagation of pathological routing update information, preventing from the sustained routing oscillations. It is obtained by associating a route penalty with every prefix announced from each Border Gateway Protocol (BGP) neighbor. Route penalties increase by some fixed value which depends on the types of routing updates received by the router. For instance, when a router repeatedly receives withdrawals and re-announcement messages about the same network prefix from its neighbors, the penalty about the route to that prefix will increase. When the penalty exceeds the suppression threshold of the router (the value of the threshold depends on the manufactory and can be modified by the administrators) the route will be suppressed. The penalty also exponentially decays with time, so that the routes will not be penalized forever. It is suggested that a much simpler scheme can count the number of flaps (ie, the changing of route because of router's misbehavior and oscillations) associated with each (prefix, peer) pair during a certain time period. This kind of measurement reflects the stability of a route. Then, a router uses locally configured thresholds to decide when a route will be suppressed and when the suppressed route will be reused and re-advertised.

On the other hand, CIDR tries to reduce the number of routing entries and flaps by a simple mechanism. It removes the boundaries among address classes A, B and C, and allows bounds to be at any bit. It integrates a series of small network prefixes into a single route prefix that seems to be a large network. Because the aggregation could reduce the number of networks visible on Internet and damp route flaps by

absorbing the flaps within a larger aggregate. Hence, it can measurably attenuate the routing instability problem[Hal97].

Because of the extremely fast growth in size (over 394 million hosts on Internet by January 2006), even using RFD and CIDR are used, Internet routing tables are still very large, and the overhead of routers and the routing instability problem are still very heavy.

Slow Convergence To ensure route convergence, the routers keep exchanging information of reach-ability and try to maximize their local route preference which is used to compute the routes. The process continues until all routers agree on a stable set of routes. Slow Convergence occurs when the process takes a very long time as compared with the average end to end delay. Slow convergence has been known as the characteristic of distance vector routing (DV) algorithms[WD88]. DV routing requires that each node maintains the distance from itself to each possible destination and the vector, or neighbor, which is used to reach that destination. Whenever this connectivity information changes, the router transmits its new distance vector to all neighbors, allowing each one to recalculate its routing table. DV routing can spend a long time on the converge after a topological change because the routers do not have sufficient information to determine whether their choice of next hop will cause routing loops[LABJ00]. The entire Internet routing between ASs is based on BGP, which is a path vector routing algorithm. Path vector routing algorithms are derived from DV algorithms and keep the information on the whole path corresponding to a route. It allows to avoid routing loops and to reduce the unnecessary propagation of the duplicate routing updates, however it does not solve the intrinsic shortcomings of distance vector algorithms. Meanwhile, the instable routes also delay the network convergence time.

Scalability Issues Scalability of routing protocols is evaluated by the router and link resource consumption. The increase of Internet size, causes the scalability problems due to the routing processing overheads like (1) CPU consumption in BGP session establishment, route selection, routing information processing, and handling of routing updates; (2) The much higher router memory requirements to install the routes and multiple paths associated with the routes, and even when some more complex hardware can be used to produce high level routers, the payload is still limited[Hal97].

2.1.2 QoS Guarantees for Multimedia Services

A number of transport protocols based on IP family are currently used in Internet. Besides TCP, *User Datagram Protocol* (UDP) [Pos80] and *Real-time Transport Protocol* (RTP) [Ahm06] are commonly deployed to provide multimedia services

in interactive application. As we know, TCP uses ACKs from the destination to guarantee the packets delivery. UDP and RTP packets, differently from TCP, are one-way and their delivery is unreliable. due to no QoS guarantee of avoiding large delay time, jitter, and packets loss. Moreover, because waiting for the feedback ACKs from the destination is too slow, especially for long distances, for the realtime services, other things must be utilized to provide QoS. To overcome the problem, different architectures have been proposed. The characteristics of two proposals, *Integrated Services*[BCS94] and *Differentiated Services*[BBC⁺98], will be described, which try to support IP QoS guarantees. As we will see, each of them still has some serious drawbacks.

Integrated Services IS architecture and QoS model try to provide IP application with end-to-end 'hard' QoS guarantees. The application may explicitly specify its QoS requirements which will be guaranteed and met by the network. The Resource Reservation Protocol (RSVP) is used here to signal the resource requirements of the application to the routers situated on the transit path between the source and the destination[BZB⁺97]. Unfortunately RSVP is not suited for deployment on high bandwidth backbones due to its reliance on per-flow state and per-flow processing. The major drawback of Integrated Services is that the amount of the state information, which is required to be maintained per node, is proportional to the number of application flows, and does not scale[BCS94].

Differentiated Services DS architecture has been proposed to overcome the perceived limitations of IS. DS allows IP traffic to be classified into a finite number of priority and/or delay classes. At congested routers, the aggregate of traffic flows with a higher class of priority have a higher probability of getting through. Traffic with a marked delay priority is scheduled for transmission before traffic is less delay-sensitive. The DS architecture is composed of a number of functional elements, namely packet classifiers, traffic conditioners and per-hop forwarding behavior (PHB). The PHB describes the externally-observable forwarding behavior of a DS node. Each service class is associated with a PHB. However, it is not necessary for all elements to be present in all DS-compliant node. These elements are normally placed in ingress and egress boundary nodes of a differentiated services domain and in interior DS-compliant nodes. Since the entire Internet structure is complicated, where is the optimal place to put the DS elements is a complicated task[BBC⁺98, NBBB98].

2.1.3 IP Multicast

For group application, where more than two users are exchanging messages and maintaining the shared state, the number of unicast connections increases rapidly as the number of users increases. In order to prevent application from needing to

know about all users in the group, to be responsible for maintaining all these connection, and to decrease network load, the multicast is required. The multicast is the efficient emulation of a broadcast service for the interested users within the constraints of a network environment[Qui01].

IP multicasting protocols rely on the Internet Group Management Protocol (IGMP) to manage multicast groups. IGMP allocates the reserved Class D IP addresses to groups. There is a one-to-one relation between a multicast group and a D address at any given time[Fen97]. The addressing abstraction is used in the multicast routing protocols. The IP multicast routing protocols can themselves be divided into two basic sets of groups, i.e. *Source-based multicast trees* and *Core-based multicast trees*. Both groups are used within a single managed network or domain. They are not suitable for the overall Internet. The exchanging of information between a multicast source and other domains enable the branches of spanning trees crossing multiple administrative domains to be established. It is another entirely different problem.

2.1.4 Mobility Support of Internet Access

With the development of wireless technology, there is a big stress to have Internet access everywhere around the world not only for the military force but also for commercial services, education, health and scientific research[AJ97, HL01]. Nowadays, it is possible to use some wireless communication base stations in the cities to provide some Internet data services with low bit rate for wireless terminals. However, in suburban areas, deserts, huge mountains, steamships in the oceans, there are still the large uncovered areas that do not have Internet access at all.

2.2 Background: Satellites Constellations

Satellite-based networking has developed in complexity over the years. When the utilization of networking with satellites began, the individual satellites in geostationary orbit (GEO) was used, in which the uplink signals were amplified, frequency-shifted and broadcast were down to a large ground area with the use of the simple transparent bent-pipe repeaters onboard the satellites. the used orbits are lower than the geostationary orbit in Low-earth orbiting (LEO) and medium-earth-orbiting (MEO) satellite constellations, which have been proposed to use for the coverage of the Earth so that the ubiquitous access can be well supported. Figure 2.1 shows the orbital altitudes for different satellite constellations[Gav97, Woo01a].

GEO Satellites are located at an altitude of 35,786km above the Equator, the angular velocity of a satellite in the orbit matches the angular rate of rotation of the

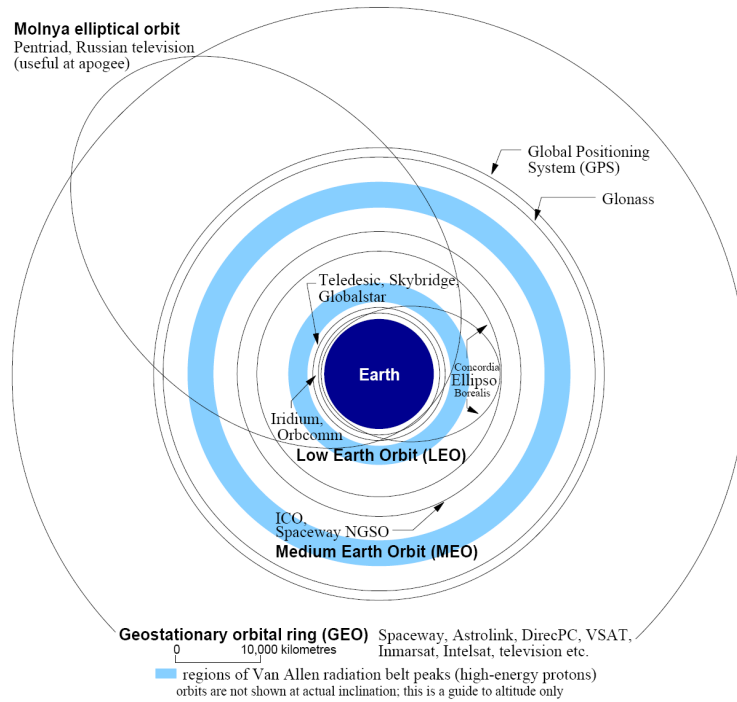


Figure 2.1: Orbits of different Constellations

Earth's surface. It makes the satellite show stationary to an observer on the Earth. However, it is impossible to cover the high latitudes (≥ 81 latitude) and rare to cover the latitudes between 75 and 81 latitudes, so full Earth coverage cannot be achieved using any purely geostationary constellation. However, the most of the Earth can be covered with a minimum of three geostationary satellites. The propagation delay between an earth station and a geostationary satellite varies with the difference in position in longitude and terminal latitude, and is around 125ms (milliseconds), but the delay is around 250ms between ground stations. It leads to the widely quoted half-second round-trip latency for communications via GEO satellites[Gav97].

MEO Satellites The orbits at the altitudes between 9,000 and 11,000km (between the inner and outer Van Allen belts) permit the fewer and smaller satellites to cover the full Earth. The satellites at lower altitudes not only have smaller coverage footprints, but also decrease the resulting delay, as compared with GEO satellites. Not like GEO's fix position, MEO satellites have the slow movement. However, MEO satellites do not remain stationary anymore and each one has visibility times of tens of minutes before the handover must take place. The propagation delay for the uplink or downlink between earth station and satellite is typically under 40ms.

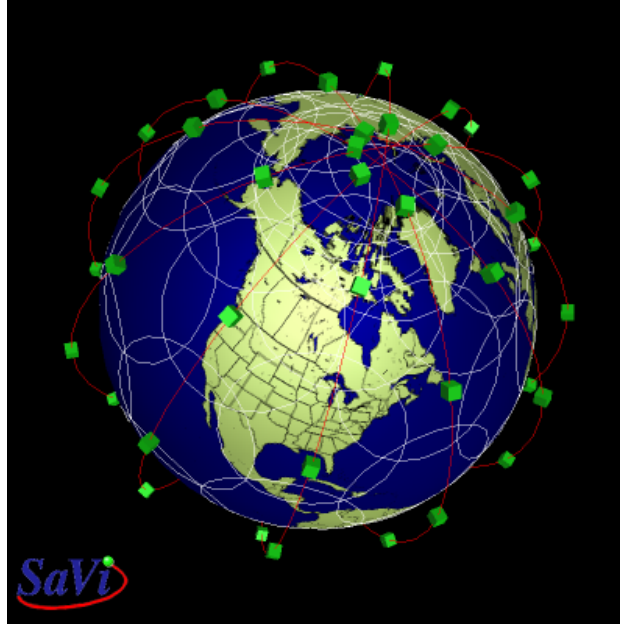


Figure 2.2: Global Coverage from Iridium Constellation

LEO Satellites At the typical altitudes between 500 and 2000km, (from the peaks of the inner Van Allen radiation belt to the upper atmosphere), a large number of satellites are required to provide simultaneous global coverage in the low earth orbit. The used actual number of satellites depends upon the required coverage required and upon the desired minimum elevation angle desired for communication. Because of much more satellites and their resulting small footprint areas (the Iridium design in Figure 2.2) and small spotbeam coverage areas, a large amounts of frequency reuse become possible across the Earth, providing large system capacity.

LEO satellites move rapidly relative to the surface of the Earth and to the ground terminals that they communicate with. When the speeds of LEO satellites are over 25,000 km/hour, the visibility is only a few minutes before the handover to other satellite occurs, the speeds are the norm. The propagation delay between ground and LEO is often under 15ms, and varies rapidly as the satellite approaches and leaves local zenith, passing the ground terminals. As compared with GEO and MEO, LEO has extraordinarily shorter propagation delay between the earth station and the satellite. But LEO satellite has a fast movement which is relative to the earth, thus the LEO constellation remains in a dynamic topology.

Table 2.1 summarizes the main characteristics of different satellite constellations:

Orbit type	LEO	MEO	GEO
Orbit Characteristics			
Altitude (km)	700-1400	10,000-15,000	36,000
N. Satellites for global coverage	40+	10-15	3-4
Orbital Period	100 minutes	6 hours	stationary
Visibility of a satellite	short	medium	mostly always
Link Characteristics			
Propagation delay	short (0.05 s)	medium (0.10 s)	long (0.25 s)
Propagation loss	low	medium	high
Elevation Angle	low	medium to high	low to medium
Call Handover	frequent	infrequent	never
Operations	complex	medium	simple
Building penetration	poor	poor	none
Ground equipment			
Battery session length	long	short	very short
Battery weight	low	high	very high
Set weight	<1/2 Kg	≈10 Kg	heavy
Set size	shirt pocket	suitcase	table
Set deployment time	immediate	15-30 minutes	hours
Satellite Characteristics			
Space Segment Cost Satellite	Highest	Lowest	Medium
Satellite lifetime (years)	3-7	10-15	10-15
Telephony Services			
Terrestrial Gateway costs	high	medium	low
Hand-held terminal possible	yes	yes	difficult
Hand-held terminal costs	low	low	medium
User density	high	medium	low
User mobility	high	semi-mobile	stationary
Broadband Data Services			
Store and Forward possible	yes	not required	not required
P2P connections possible	no	no	yes
Broadcast Services			
Service Area	small	large	very large
Antenna type	small	self-adjusting	fixed and large
System reliability			
Single satellite failure	high reliable	somewhat reliable	low reliability
Multiple failures	reliable	low reliability	disaster
Network complexity	high	medium	low

Table 2.1: Comparison of LEO/MEO/GEO satellites [Gav97].

2.3 Motivation of using LEO constellations

The most extraordinary merit of LEO is the very short transmission delay time as compared with GEO and MEO. The short delay time can be adapted to the common requests of all kinds of Internet traffic. GEO and MEO satellites can not guarantee the QoS for real-time service because of their long delay caused by the high altitude. Moreover, the LEO satellites' movement relative to the ground is rapid due to low altitude. The high mobility leads to a rapid and regular change of network topology, and raises numerous issues for the networking layer with respect to routing. However, the topology of LEO constellations exhibits interesting and useful properties[Woo01b]:

- Predictability - the movement of satellites are predictable.
- Periodicity in the space segment - satellites periodically appear in the space.
- Regularity - the dynamic topology is regular.
- Constant number of satellite network nodes.

From the point of view of the Internet, LEO constellations also exhibit good chances to overcome the mentioned problems above.

Routing with LEO Constellation We want to use LEO satellites constellations to extend the multilayer routing structure of Internet. The LEO satellite constellations can be regard as a special autonomous system which can provide worldwide access. It is hopefully for us to use LEO for establishing the shorter AS paths from sources to destinations, especially for long distance access, as compared with ground Internet infrastructure. Also it is helpful to overcome slow convergence time and scalability problems. Furthermore, if a LEO constellation serves as an autonomous system, the routing policies used inside the constellation are hidden in the hosts on ground. It is possible for us to exploit some new routing policies in the LEO satellite system without special requirements to modify the terrestrial routing architecture.

QoS with LEO Constellation As mentioned above, the constellation can be considered as a single AS, in which new QoS policies based on the characteristic of spectrum and frequency can be used for real-time and multimedia traffic aggregate. At the same time, the usage of LEO constellation can reduce the terrestrial ASs so that the real-time data streams will traverse, reducing the influence of ground network congestion and providing high QoS. Finally, since the mechanism used in the constellation has no influence on the ground Internet, some kinds of admission control schemes can be utilized for high QoS.

Multicast with LEO Constellation Because of LEO satellite's properties, e.g. limited number of satellites, simple structure and worldwide coverage, it is possible for LEO satellite to use inter satellite links for building a multicast tree of redistribution to all communicating ground parties involved in the remote terrestrial terminals. In particular, the special transmission medium (spectrum) and transmission technology (broadcast), which are used by LEO satellites to communicate with ground stations, can naturally bring multicast into realization.

Mobility supported with LEO constellation LEO satellite constellation's global reach-ability and wireless transport medium are expected to have a profound effect on the organization and individual operators who want the ubiquitous Internet services. For instance, with LEO satellites support, the scientific research groups of the deep sea can directly send the collected data to the labs in the universities for analysis; doctors, not in their offices, can diagnose the diseases for the patients by downloading the medical examination reports of the patients from the Internet. Moreover, the outer space scientific research, like Moon and Mars explorations, also need satellites to support data, voice and video services for data collection and remote equipment control.[EAB01, AAC⁺03]

2.4 Technical challenges of integrating Internet with LEO constellations

In this section, the aspects of the integration architecture will be analyzed and the main technical challenges addressed in the thesis will be outlined.

2.4.1 Multiple Access Control

In Multiple Access Control (MAC) protocols, a set of rules for controlling access to a shared channel among contending users are defined. MAC protocols play an important role in efficiently and fairly utilizing the limited satellite system resources as they can significantly affect higher-layer protocols and the QoS provided by the system. Generally, the MAC should be able to achieve high throughput, maintain channel stability, and enjoy low protocol overhead and small access delay. The MAC protocols in the LEO constellations should also have the following properties:

- should provide a scheduling algorithm for many different user terminals or servers to contend for satellite up-link channel in the same footprint (due to smaller terminals VSAT/USAT);
- should allow of the maximum utilization of available bandwidth;
- should allow the predictability for the QoS provided by the system (eg, on delay, on delay jitter, on packet loss ratio etc);

- should provide the different classes of service (eg through priorities).

According to the bandwidth assignment, MAC schemes for LEO satellite constellations can be classified into three groups: fixed assignment, random access, and demand assignment.

Fixed Assignment – Fixed assignment may be on the basis of frequency, time or code. Major techniques include frequency-division multiple access (FDMA), time-division multiple access (TDMA), and code-division multiple access (CDMA). FDMA and TDMA systems allow of the predictability and free contention. They can provide QoS guarantee. However, these protocols intrinsically lead to inefficient medium utilization due to the empty slots devoted to non communicating entities.[Mar05] On the contrary, in CDMA, each communication is assigned to a unique code sequence which is used to spread the data signal over a wider bandwidth than that required to transmit the data. If code sequences are guaranteed to be orthogonal, all other simultaneous transmissions in the same channel act as additive interference to the desired signal and can be removed completely at the receiver side to recover the original data. Thus, CDMA system can make the full use of the bandwidth and make it more flexible for system expansion.[Mar05]

Random Assignment – In random access schemes, each station transmits the data regardless of the transmission rate of the others (i.e., Aloha[KM99]). The retransmissions after collisions increase the average packet delay, and the frequent collisions may cause low throughput. Other disadvantage of random assignment is the unpredictability of transmission, which can not provide serious QoS guarantee.

Demand Assignment – Demand Assignment Multiple Access (DAMA) protocols dynamically allocate the system bandwidth in response to user requests[Mar05]. Before actual data transmission, a resource request must be granted. After a successful reservation, the bandwidth is allocated on a FDMA or TDMA architecture, and the transmission is collision free. The reservation may be made under centralized or distributed schemes/protocols. The central controller can be located at ground station or at a satellite with onboard processing. While in the distributed control mode, each ground station receives all requests from the satellite broadcast channel and makes a decision locally. The distributed control is more robust and reliable. The overhead of reservation announcement is reduced greatly and the minimum reservation delay is as small as one round-trip time. The reservation may be explicit or implicit. The explicit reservation is on the basis of a clear reservation request. Each station sends a short request specifying the number of slot needed by the transmission. In implicit reservation, there is no request message. The packets belonging to a long transmission can repeatedly occupy the same slot in consecutive frames.

An empty slot in a frame indicates the end of the transmission, and other users can then contend for this slot in the next frame.

2.4.2 Routing

Routing issues are crucial for LEO constellations with intersatellite links (ISLs) and onboard processing (OBP). Routing should allow of the maximum utilization of available bandwidth, balancing load among satellites, and keep low (maybe predictable) delay to provide high QoS. As mentioned above, the major technical challenge is routing in a complex dynamic context due to the satellite movements. Dynamic in the satellite constellations is twofold:

- *Inter-Satellite user handover* : Considering the relative movement between the LEO constellation and the Earth, a satellite has a very short lifetime to fixed users on the ground (around 15 min per satellite). When a satellite moves out, the users must switch to the next satellite coming in their sight.
- *Inter-plane ISL switch off*: The ISLs form a mesh network topology inside the constellation[Gav97, HL01]. There are two types of ISLs: intraplane ISL connecting adjacent satellites in the same orbit and interplane ISL connecting neighboring satellites in adjoining orbits. Intraplane ISL can be maintained permanently, while interplane ISL may be temporarily switched off when the viewing angle among two satellites changes too fast for the steerable antennas to follow. It may happen when two orbits cross, such as near the pole.

The dynamic routing algorithms that are used in ground Internet, like Distant Vector (DV) and Link State (LS) algorithm, can not be directly applied in LEO constellations, because they can not adapt to the frequent topological changes[Woo01b]. If these mechanisms are used, they will cause large overhead and route oscillations. Two new concepts of routing topologies for the satellite constellations have been proposed: Discrete-Time Dynamic Virtual Topology Routing (DT-DVTR)[Wer97, HL01] and Virtual Node (VN)[Woo01b, HL01, MR97].

DT-DVTR – DT-DVTR uses fully the periodic nature of the satellite constellations and works completely offline. It divides the system period into a set of time intervals so that the topology changes only at the beginning of each time interval and remains constantly until the next time interval. In each interval, the routing problem is a static topology routing problem that can be solved much easily. Because of offline computation, a large storage on board is needed for the routing table which compensates for reduced on-line computation complexity. Even if an optimization procedure can be used to choose the best path (or a small set of paths) from the series of routes to reduce the storage size, some links may become congested while others are underutilized.

VN – The VN concept tries to hide the mobility of the satellites from the routing protocols. A virtual topology is set up with VNs superimposed on the physical topology of the satellite constellations. Even if the satellites are moving across the sky, the virtual topology remains unchanged. The states, such as routing table entries and channel allocation information are transferred continuously from one satellite to other. Routing is performed in the fixed virtual network using other routing protocols that are designed for satellite constellations.

According to the two new topology concepts discussed above, the routing schemes have been proposed on the basis of IP[HS98, EAB01, NCD98, CEA02] and ATM switching[HMK98, Wer97, WDV⁺97].

However, both classes have some unsolved problems:

- Problems with IP routing in LEO Constellations
 - *Variable length IP packets*: As the satellite air interface must allocate the channel capacity in some predefined manner via FDMA/TDMA, the fixed datagram sizes are needed to fit neatly into the frame structures for the allocated slots in the wireless channel. This poses the problems with IP variable-length packets. The author proposes to fit IP packets into any fixed length by breaking up IP in order to be carried by a MAC-level or a tunnelling protocol [Woo01b]. However, the performance of such strategy has not been analyzed and no simulation is discussed, which convinces the reader that this strategy solves the problem smoothly;
 - *Limited memory for onboard routing table management*: The satellite computing performance is expected to lag behind equivalent terrestrial performance at any point of time. Once a satellite is launched it can not be upgraded for the duration of its expected using time. So this is a clear argument for placing the routing complexity in the ground segment of the constellation network as much as possible. To a certain extent, this argument also aggravates the following problems.
 - *Requirement for large computational capacity*: Although almost all the satellites do not need participate in entire routes computing from sources to destinations, they still should calculate the route to forward packets inside the constellation AS, including the ground gateways. Also rerouting is needed after congestion happens in part of constellation.
 - *Load Unbalance*: The different areas around the Earth have different densities of gateways. Thousands of gateways may be deployed in one satellite footprint above the big cities, while in the deserts without anybody there may be no gateway to be connected with the constellation at all. This will cause the load unbalance in the constellation. The satellites above some areas may be always busy in routing and forwarding the packets, while others are idle.

- Problems with ATM switching in LEO Constellations
 - When ATM is used as the network protocol for the constellation with DT-DVTR [Wer97] to provide Internet service, IP over ATM or other similar technologies should be utilized. IP packets from the Internet should be split into ATM cells by the base station and reunited after they pass through the constellation infrastructure[AJ97]. However, besides those problems caused by IP, IP over ATM has the drawback about congestion control. ATM usually uses an available bit-rate (ABR) virtual channel to tunnel the TCP/IP packets across the constellation network. ABR is generally closed-loop to communicate its congestion state back the endpoints so that the output can be adjusted. But TCP takes open-loop approach. If the ATM network is congested, there is no way that ABR with explicit rate feedback can tell the TCP endpoints how much output they need to decrease. TCP endpoints do not react to the congestion, thus the congestion problem can not be solved[Woo01a].

Another problem for implementing routing in constellations is related to the hardware of satellites. If the heterogeneous routing inside the constellation is employed, because of the different satellite manufacturers, LEO constellations can be considered as an Autonomous System with a set of boarder gateways (BGs) running exterior routing protocols (BGP or one extension of BGP) that communicate with terrestrial ASs. Only BGs need to know the topology and addresses outside the constellation. BGs are the entries/exits for those packets that want to pass through the constellation. At the same time, BGs perform encapsulation/decapsulation and address resolution. The BGs can be implemented either on board of the satellites or in ground GSs: Moreover, both solutions have some drawbacks:

- a space-based BGs– The computation and storage requirements may be too much for the satellites.
- b terrestrial gateways– The packets must be bounced back to the ground for IP routing, introducing an extra round-trip delay.

2.5 Summary

In this chapter, we have explain the problem of Internet and the reason that we want to use LEO satellite as a solution. In the second part we focus on the technical challenges of using LEO satellites to provide Internet services. In chapter 3, we will talk about the details of physical layer and medium access in LEO constellations.

Chapter 3

Satellite Links and Protocols

In this chapter, we will briefly discuss all the protocol stack used in LEO satellite constellation from the lower physical level. We will discuss LEO constellations both with and without links between satellites to see the importance of using intersatellite links. After that we will examine the main MAC candidates for intersatellite links and their properties. Finally, we will focus on the transport layer problem of LEO satellite constellations. Networking layer, which is the main topic of this thesis, will be examined in detail in Chapter 4.

3.1 Constellations and Intersatellite links

3.1.1 Intersatellite links

Intersatellite links, or ISLs, are those connections that are used by the satellites to form the satellite constellation network over the surface of the planet. Ideally, if all satellites are identical, and able to maintain the same number of ISLs (degree of connectivity), then we have a regular network mesh composed of intra-plane and interplane ISLs[Woo01a].

Intra-plane ISLs are those connections between satellites within the same orbital plane, (Figure 3.1). They are generally permanent if the orbits are circular, as the satellites positions remain fixed relative to each other[Woo01a].

Interplane ISLs are links between satellites in different orbital planes. These may not be permanent. Because neighboring orbits cross each other near the highest latitudes, where each satellite neighbors will swap sides, this requires terminals between ISLs to either physically slew through 180° to follow the neighbor and maintain the link, or requires that the links be broken and remade[Woo01a].

ISLs can be classical, using radio frequency (RF) (like in Iridium [LMG93, JP93])

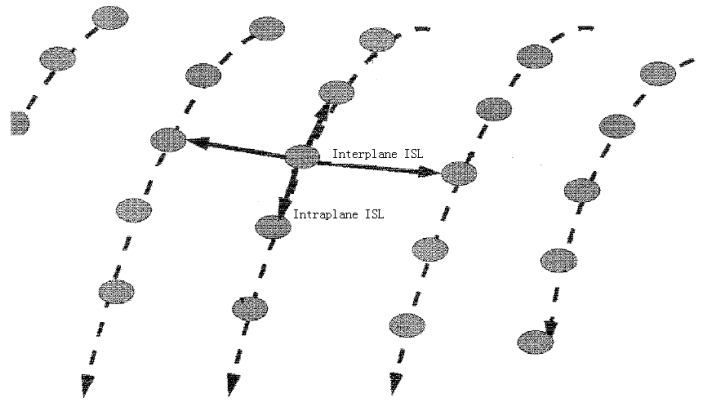


Figure 3.1: Interplane and Intraplane ISLs

or optical (like in Teledesic [SG96]). The optical links seem to be more promising because they can offer high link capacities and on the other hand reduce size, weight, power and, of course, the cost of the equipment on board of the satellites[SG99]. With respects to the RF links, optical links could potentially provide tens of dBs (unit of wastage rate) of link efficiency improvement, eg., data rate, etc, (We will discuss this problem in more detail in Section 3.3). However, optical systems typically have much narrower beamwidth than RF systems. The advantage of a narrower beamwidth is that the potential for interference to or from adjacent satellites will be reduced (this is very important in large LEO constellations). The disadvantage is the need for more accurate pointing, acquisition and tracking and the impact that this may have on the spacecraft as it could impose an unwelcome burden. More comparisons between RF and optical will be discussed in Section 3.3.

Lots of technical issues should be discussed on using ISLs such as coverage, delay, handover, link feasibility, power and cost restrictions (link budget) due to the choice of RF or optical ISLs. The advantages and disadvantages of ISLs are resumed in [Örs98], as shown in Table 3.1. First of all constellations with ISLs greatly reduce the propagation delay of transmission, but on the other hand, the onboard processing consumes more power. Secondly, ISL constellations can provide more coverage area than non-ISL constellations in areas without ground base stations like oceans and deserts. Moreover, non-ISL constellations need more network control units and frequency coordination. Control units manage and allocate the limited frequency to transmissions between ground and space segments, especially for long distance multi ground-space hop connections.

ADVANTAGES	DISADVANTAGES
calls may be grounded at the optimal ground station through another satellite for call termination, reducing the length of the terrestrial tail requires	complexity and cost of the satellites will be increased
a reduction in ground-based control which may be achieved with on-board baseband switching - reducing delay (autonomous operation)	power available for the satellite/user link may be reduced
increased global coverage (oceans and areas without ground stations)	handover between satellites due to inter-satellite dynamics will have to be incorporated
single network control centre and earth station	replenishment strategy
	frequency co-ordination
	cross-link dimensioning

Table 3.1: Advantages and disadvantages of using ISL [Örs98]

3.1.2 Constellation categories with respect to ISL

Constellations can be classified in two categories depending on using ISL or not. Non-ISL LEO satellite systems, which usually employ bent-pipe transponders, as shown in Figure 3.2, are used only to provide connectivity between terrestrial gateways. This kind of constellation is usually composed of satellites without onboard processing and ISLs, for instance SkyBridge system [SO97, FCM00]. Because main processing units are all on the ground, we call this type of constellation *Ground-based*. A schema for this architecture is defined in Figure 3.3, each satellite has a footprint on ground and numbers of gateways are deployed inside each footprint. Packet routing is performed through the ground network: the source node transmits its packet to the satellite, which forwards it to the gateway on its footprint. Then, the packet travels through the ground network up to the gateway of the footprint of the receiver, which in turn transmits the packet to the satellite of the destination. The route of a transmission is composed by a series of earth-space-earth connections. In brief, non-ISL satellites have low costs compared to ISLs satellites but they need a lot of joint work from the terrestrial network.

On the other hand, a constellation with ISLs, named *Space-based constellation*, provides direct connectivity between satellites by using radio or optical intersatellite links to reach the satellite covering the final destination. Non-ISL constellations connect a pair of remote server/user by using a sequence of satellite/gateway connections, and this will reduce efficiency of the limited radio frequency. Using ISL can reduce the space-earth-space delay and provide radio frequency re-usage, by decreasing the amount of earth-space-earth traffic needed to connect distant ground

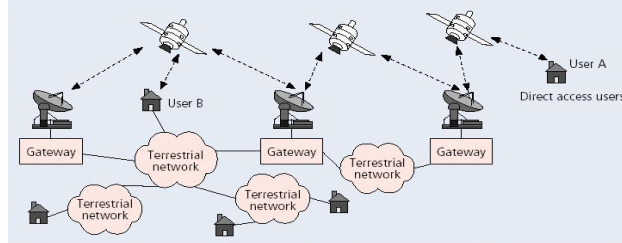


Figure 3.2: Ground-based constellation [HL01]

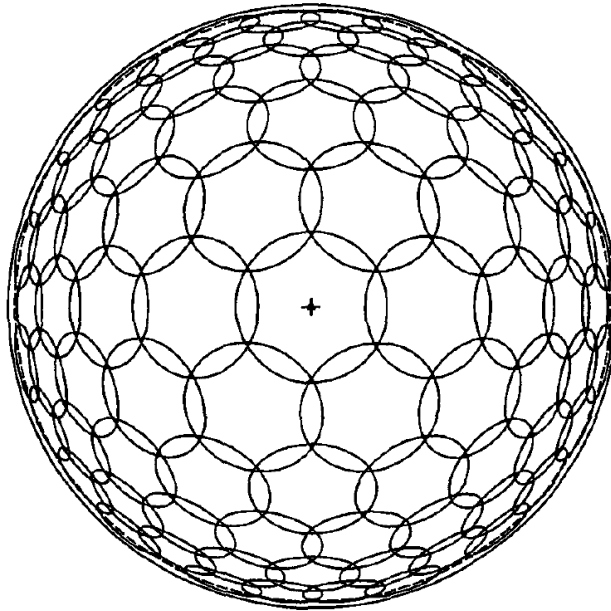


Figure 3.3: Footprint of SkyBridge System with 80 satellites [FCM00]

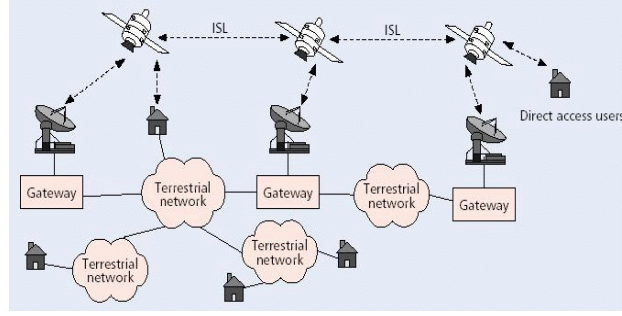


Figure 3.4: Space-based constellation with ISLs [HL01]

terminals by means of ground gateways, as happens in ground-based constellations. Figure 3.4 shows the typical scenario of a space-based constellation. Satellite constellations with ISLs are true networks, so they must support on-board switching and on-board routing[Woo01a]. On-board processing and routing will be discussed in Subsection 3.1.3.

ISL organization makes the space segment of the constellation more complex and sophisticated and it allows the provision of ubiquitous services in regions where it is difficult to place a gateway station.

On the other hand, ground-based constellation networks separate the network functionality from the space segment, so allowing the reuse of the satellites for different purposes by simply changing the ground segment functionality. By contrast, in space-based networks, networking and space segment issues must be deployed together.

3.1.3 Onboard switching

Usually, satellites with ISLs allow on-board processing, including decoding/recoding, transponder/beam switching, and routing to provide more efficient channel utilization.

On-board switching can be achieved basically with two types of switch: *circuit switch*, and *packet switch*[Ngu03]. Circuit switching adopted by on-board processing transponders is based on a deterministic technique, i.e. the configuration of the system is defined at the beginning of the connection, and it goes on for the entire duration of a connection. The main advantages are that, since no overhead is needed for switching, high throughput is achieved, and, since resources are statically allocated, loss rate is zero. On the other hand, circuit switch is not very flexible, since

it assumes predictable traffic, and consequently it is not very reactive. When the circuit switched network is used to support packet-based traffic (like IP), or, more in general, short-lived connections (like messaging), the setup delay may represent a large part of the total connection time, thus reducing the network's capacity. In addition, reserved resources cannot be used by any other users even if the circuit is inactive, and this may reduce link utilization. Thus, circuit switches are not very suitable for broadband services[Mar05].

In the more popular packet switching technique each message is split in equal length packets, each one containing the address of origin, the address of its destination, and other control information (i.e. about how to merge it with other related packets). Packets are treated independently of each other: so it is more dynamic, and more adaptable to unpredictable traffic. Contention situations are solved instantaneously, according to the information present in packet headers and thanks to on-board buffers. However, buffers, where incoming packets are queued before they are sent out on outgoing links, imply that if the rate at which packets arrive at a switch exceeds the rate at which packets can be transmitted, the queues grow and buffer overflow can occur causing packet loss. This may happen, for example, if packets from several incoming links have the same destination link. Loss of data usually causes retransmissions that may either add to the congestion or result in a less-effective utilization of the network. Moreover, queuing causes delay, and a very good buffer handling to support real-time traffic is needed[Mar05].

Table 3.2 summarizes the characteristics of these two on-board switching technologies. Generally, the circuit switch has a lower bandwidth utilization than the packet switch. On the other hand, packet switch has high overheads for circuit traffic due to the packet headers, and contention/congestion may occur when using the packet switch.

3.2 Satellite Protocol Stack Overview

In this section, we will briefly outline the protocol stack used in LEO satellite constellations.

Differently from Internet's 100/1000 Mb cable connection, LEO constellation has Radio Frequency or Optical based physical layer. Meanwhile, optical constellations require accurate pointing and tracking techniques which are more complex than RF constellations. And for RF constellation, special MAC techniques like FDMA/TDMA/CDMA should be used in order to schedule data packets in LEO constellations. Difference between optical and RF constellations will be discussed later in Section 3.3.

	Advantages	Disadvantages
Circuit switch	Excellent solution for circuit-based service provisioning	reconfiguration of earth station time/frequency plans for each circuit set-up
	easy congestion control by limiting access into the network	no fixed bandwidth assignment
		low bandwidth utilization
		difficulty of implementing autonomous private networks
Packet switch	self-routing and auto-configuration abilities	for circuit switched traffic, higher overhead is required than circuit switch due to packet headers
	flexible and efficient bandwidth utilization	contention and/or congestion may occur
	can accommodate circuit-switched traffic	
	easy to implement autonomous private network	

Table 3.2: Comparison of different on-board switching technologies [Ngu03].

For network layer issues, traditional routing protocols in IP networks like RIP/OSPF have big limitations due to LEO's propagation delay and the high link error rate. Routing instability problems can be severe. There are two different solutions to routing in LEO constellation. One is to use IP tunnelling or NAT techniques to implement IP [Woo01b]. The other is to use private routing protocols for LEO constellations. In the latter case, different routing algorithms should be designed. In Chapter 4, we analyze the network layer problems.

The transport layer protocols that are used in traditional Internet like TCP also have their own limitations. Traditional TCP can not distinguish data loss caused by congestions and link errors. LEO satellites' propagation delay also increases the time cost of initial window and recovery. TCP mutations are considered to be used in LEO constellations [RF99, CGM⁺01, AMP01]. These mutations modify TCP to be more adaptive for LEO constellation. Another solution is to use explicit TCP proxies between terrestrial and space segments. New techniques concerning transport layer in LEO constellation will be discussed later in section 3.4.

Figure 3.5 illustrates the difference in the protocol stacks between terrestrial Internet and LEO satellite constellations. In the rest of this chapter, we will analyze

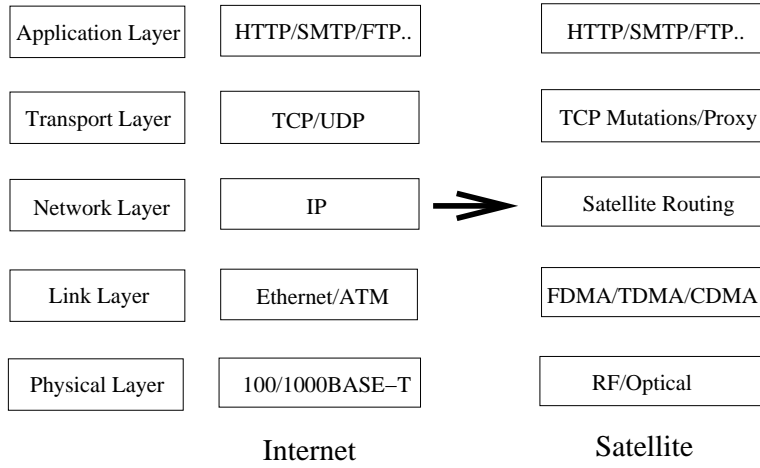


Figure 3.5: Satellite Protocol Stack vs. Typical Internet Stack

the problems related to the MAC layer and the transport layer.

3.3 MAC Layer

3.3.1 Optical vs. Radio Frequency

As two main intersatellite medias, radio frequency and optical links have their own properties and advantages. Optical links, which are normally implemented by using laser, have high data rates and low power. But the whole concept of laser ISLs is based on the precise pointing of the transmitting beam towards the receiving satellite. This requires a very precise estimation of the position and synchronization of the satellites. Research has been proposed to address laser ISL problems [LC00]. The problem of the precise pointing of the transmitting beam is regarded as a position estimation problem. The position estimation is related to propagation delay measurement and ranges from the source satellite to others. But the computation of transmitting beam acquisition and satellite tracking are still complex, especially for LEO constellation with large satellite numbers.

On the other hand, although radio frequency has a lower data rate than laser, it is still on the order of the requirement of usual ISLs, which is about 1Mbps. RF uses directional antennas for data transmitting. It is not as precise as laser ISLs. Radio frequency is more flexible and reliable than laser optical ISLs. The main problem of using radio frequency is solving efficiently the multiple access problem. There is research about multiple access control for wireless networking, including satellite environments. This will be discussed in detail in the subsection 3.3.2.

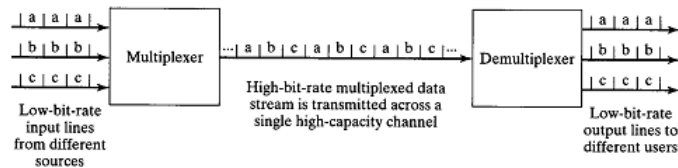


Figure 3.6: Time Division Multiplexing [SM06c]

3.3.2 Multiplexing techniques

The switching of data on board a satellite can be classified into different types of multiplexing techniques:

Time Division Multiplexing (TDM) TDM is a type of multiplexing that combines a set of low-bit-rate data streams into a single high-speed bit stream that can be transmitted over a single channel by assigning each stream a different time slot, as shown in Figure 3.6. The main reason for using TDM is to take advantage of existing transmission circuit lines. If each low-bit-rate transmission occupies one costly channel then lots of network resources will be wasted. TDM is commonly used for wireless telephone networks [SM06c].

Code Division Multiplexing (CDM) CDM allows signals from a series of independent sources to be transmitted at the same time over the same frequency band. This is accomplished by using orthogonal codes to spread each signal over a large, common frequency band. At the receiver, the appropriate orthogonal code is then used again to recover the particular signal intended for a particular user [SM06a].

Frequency Division Multiplexing (FDM) FDM means that the total bandwidth available to the system is divided into a series of nonoverlapping frequency sub-bands that are then assigned to each communicating source and user pair, as shown in Figure 3.7. FDM is the simultaneous transmission of multiple separate signals through the shared medium by modulating, at the input, the separate signals into separable frequency bands, and adding those results linearly. While thus combined, all the signals may be amplified, conducted, translated into frequency and routed toward a destination as a single signal. The receiver, separates the multiplexed signals by means of frequency passing or rejecting filters, and demodulates the results individually, each in the manner appropriate for the modulation scheme used for that band. [SM06b]

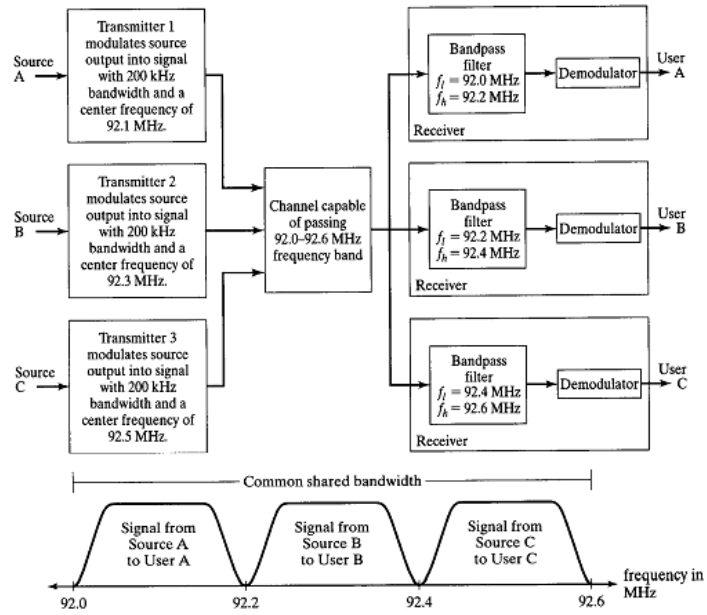


Figure 3.7: Frequency Division Multiplexing [SM06b]

3.3.3 Multiple access techniques

With the growth of the access demands to the constellation, the problem concerning the efficient use of the satellite transponders is raised. The multiple access is a technique that allocates the resource of the satellites. Based on the multiplexing techniques described in last subsection, therefore, there are three common multiple access protocols:

1. Frequency Division Multiple Access (FDMA);
2. Time Division Multiple Access (TDMA);
3. Code Division Multiple Access (CDMA).

Typically, the FDMA technique is suited to networks with few stations and a high capacity because too many stations will cause too many narrow subchannels divided by the sender/receiver pairs, whereas the TDMA technique is more suitable for networks with the opposite characteristics because lots of low-bit-rate transmission together can make full use of the entire channel. However, in many practical cases, combinations of such techniques are used as, for instance, the Multiple-Frequency TDMA (MF-TDMA).

3.3.4 Frequency Division Multiple Access

This is probably the most widely used system and the easiest to implement. In FDMA, the available transponder bandwidth (or a part of it) is divided into sub-bands of non-overlapping frequency slots, which are assigned to individual earth stations. There are two ways of implementing the FDMA technique:

- Single Channel Per Carrier (SCPC);
- Multiple Channel Per Carrier (MCPC).

According to the SCPC method, each carrier can host only one channel, while in the case of the MCPC technique, all user channels connected to a station are multiplexed so as to form a single flow of data (carrier). Furthermore, this multiplexing technique ensures that the channels destined to the various receiver stations can be separated [AJ97, Mar05].

3.3.5 Time Division Multiple Access

This technique consists in dividing the time into several intervals (called *time slots*) which are then allocated to the earth stations which transmit onto the same carrier frequency by using the whole bandwidth and communicate with each other by means of non-overlapping bursts of signals.

In TDMA systems, the transmit timing of the bursts is accurately synchronized, so that the transponder receives one burst at a time. Transmissions are organized in frames, each one consisting in a number of bursts originating from a set of earth stations in the network. Each station must be equipped with one single receiver-demodulator so that the set of data the station receives can be successfully demodulated into different individual streams [AJ97, Mar05].

Satellite-switched TDMA

A satellite-switched TDMA (SS/TDMA) system is an efficient TDMA system with multiple spot beam operation for the uplink and downlink transmissions. The interconnection between uplink and downlink beams is performed by a high-speed switch matrix on board the satellite. An SS/TDMA scheme provides a full interconnection of TDMA signals between different coverage regions by interconnecting the corresponding uplink and downlink beams at a switching time. The switch matrix is a crossbar in which only a single row is connected to a single column at time.

The advantage of SS/TDMA systems with respect to TDMA systems is the possibility of frequency re-use, since the same frequency band can be spatially re-used many times; this means that the satellite capacity increases considerably [Mar05].

3.3.6 Code Division Multiple Access

In CDMA satellite systems no bandwidth or time sharing is required. Each uplink earth station is identified by an address *code* (or signature) imposed on its carrier, and it uses the entire bandwidth to transmit through the satellite whenever desired. The receiver earth station uses the same code to reconstruct the information transmitted by the sender and to separate it from the information transmitted by the other earth stations. The whole channel is used, and the transmitted signal is the product of multiplication of the data by the transmission code for each transmitter at any given time [AJ97, Mar05].

There are two main CDMA techniques as follows:

1. *Direct sequence CDMA (DS-CDMA)*: in this technique, each source has a random sequence that identifies the transmitter and is known to the receiver. The system multiplies each binary symbol of the message to be transmitted by this sequence (or code). The receiver has a filter adapted to decode the codes it is concerned with. Codes from other stations are interpreted as random noise. The bits of the random sequence are referred to as *chips*. The ratio between the chip rate and information rate is called the *spreading factor*. Phase-shift-keying modulation schemes are commonly used for these systems. The most widely used binary random sequence is the maximum length linear feedback shift register sequence (*m*-sequence) which is generated by an *m*-stage shift register. The *m*-sequence has a period of $2^m - 1$. There are two types of DS-CDMA techniques: synchronous and asynchronous. In a synchronous system, the entire system is synchronized in such a way that the random sequence period (code period) or bit duration of all the uplink carriers in the system are in time alignment to the satellite. This requires that all stations have the same random sequence period and the same number of chips per random sequence length. Hence, a synchronous DS-CDMA must have the type of network synchronization used in a TDMA system but in a much simpler form. However, in an asynchronous DS-CDMA satellite no time alignment of the random sequence period at the satellite is required, and each uplink carrier operates independently with no overall network synchronization. Therefore, the system complexity is much simpler than a synchronous system.
2. *Frequency hopping CDMA (FH-CDMA)*: each earth station transmits for a given time period on a frequency defined by a pseudo-random law. At each time period this frequency is modified.

Figure 3.8 shows the general concepts of FDMA, TDMA and CDMA. FDMA just divides the channel into several sub-channels, while TDMA combines different transmissions into one data set and CDMA uses unique spread codes for each transmission to flexibly use network channels.

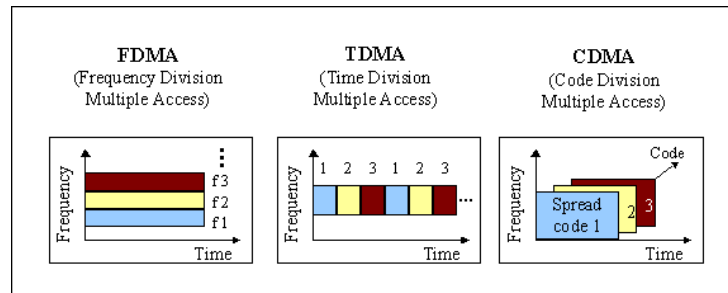


Figure 3.8: Multiple Access Technologies (<http://homepages.uel.ac.uk/u0227598/access.htm>)

3.3.7 Demand Assignment Multiple Access

Demand Assignment Multiple Access (DAMA) protocol requires a master entity responsible for bandwidth assignments. All stations, before starting transmissions, ask the master entity for bandwidth allocation. Once the master has received the bandwidth requests of all terminals, it decides how much channel resources to allocate to each terminal using an allocation algorithm. Each terminal is then informed on how much bandwidth it will receive and when to begin transmission. This multiple access technique results in a system wherein the time varying bandwidth needs of any terminal can be accommodated [AJ97].

Two main drawbacks of DAMA are bandwidth waste due to request signaling and larger packet delays. In a pure DAMA scenario, each packet must wait a round trip time (i.e. twice the time required to transmit a packet between the Earth terminal and the allocating agent) before it can begin transmission. Due to the high delay (latency) in satellite networks, passing DAMA request information between an Earth terminal and the satellite is a very slow process. The idle bandwidth during the requesting phase is wasted.

3.3.8 Random Access Protocols

For the sake of completeness, in this section we briefly describe random access protocols, such as Aloha and its variants [KM99]. We note that all these protocols can cause instability, in the sense that throughput can tend to zero and delay of packet delivery can grow indefinitely as traffic grows.

Aloha In the Aloha protocol (also named *Pure Aloha*), stations are allowed to freely access the channel whenever they have data to transmit. Because the threat of data collision exists, each station must either monitor its transmission on the rebroadcast, or await for an acknowledgment from the destination station. By

comparing the transmitted packet with the received packet, or by the lack of an acknowledgement, the transmitting station can determine the success of the transmitted packet. If the transmission was unsuccessful, it is resent after a random amount of time to reduce the probability of re-collision [KM99].

The advantages of this protocol are that it is extremely simple and suitable for a varying number of stations. The main disadvantage is that it has been theoretically proven that the maximum achievable throughput is 18.4%. Besides, it requires queueing buffers for retransmission of packets.

Slotted-Aloha By imposing a simple constraint in the transmission freedom of the individual stations, the throughput of the Aloha protocol can be doubled. In the Slotted-Aloha protocol, packets have constant length, and transmission time is broken into slots equivalent to the transmission time of a single packet. Stations are allowed to transmit only at slot boundaries. When packets collide they will overlap completely instead of partially. This has the effect of doubling the efficiency of the protocol [KM99].

If a collision occurs, the involved stations retransmit the packet at some random time, in order to reduce the possibility of recollision again. Obviously the limits imposed which drive the random retransmission of packets will have an effect on the delay associated with successful packet delivery. If the limit is too short, the probability of recollision is high. If the limit is too long, the probability of recollision lessens but there is an unnecessary delay in the retransmission.

The advantages and disadvantages of Slotted-Aloha protocol are the same as Aloha protocol, but the maximum achievable throughput is of 36.8% . Moreover, synchronization is required.

Reservation Aloha In this protocol, packets have constant length and time is divided into slots, each one equal to the transmission time of a single packet. Slots are organized in frames of equal size. Each frame is divided into two phases: one for reservation, treated with Slotted-Aloha, and one for transmission. Reservation is gained if successful transmission is made in the first phase. At the end of reservation time, all stations having successfully obtained a slot, transmit data in the same order of the reservation.

The advantages of this protocol is that it has better delay results than Slotted-Aloha. With Slotted ALOHA any slot is available for use by any station regardless of its prior usage. With Reservation ALOHA the slot is considered to be owned temporarily by the station which used it successfully. When the station is through with the slot it simply stops sending. An idle slot is available to all stations on a contention basis.[Alo06]

3.4 Transport layer in LEO constellation

We focus on the proposed work that wants to provide Internet services on top of the satellite constellation, which must support TCP/IP. In this section we first discuss the problem of implementing plain TCP over constellation (section 3.4.1). In order to overcome these problems two principal approaches have been proposed: modifying TCP and using proxies. We discuss these two approaches in section 3.4.2 and 3.4.3 respectively.

3.4.1 Implementing plain TCP

The characteristics of LEO satellite networks influence the performance of traditional TCP in several ways. One important thing is that, the large propagation delay of a constellation enlarges the time interval of TCP's "slow start". As we know, TCP senders send windows of packets. TCP controls the sending rate by adjusting the window size. The senders start to send with a window size of one segment and double this size once every round trip time (RTT) until reaching a certain threshold. Thus the increasing of the sending window size is dependent on the RTT value. Compared with physical links, LEO constellation's long RTT will worsen the performance of TCP with a very slow increasing of sending window size. Another issue that could have an impact on TCP's performance is LEO's high link error rate. The packets dropping in TCP is regarded as a phenomenon of congestion. TCP's congestion control mechanism assumes that packets discarded by the bottleneck link or overflow buffer is the main reason of packets loss. It is true for terrestrial wired networks. But in the case of wireless satellite networks, frequent link errors also may cause packet losses. Plain TCP can not distinguish whether the reason for being loss is because of congestion or link errors and thus it can not recover properly with a high performance degradation.

For these problems, two main approaches have been proposed to port TCP onto LEO satellite networks. One is carefully modifying the TCP protocol itself so that it can adapt to LEO's wireless characteristics, the other is modifying the network architecture using TCP proxy.

3.4.2 TCP Modifications

One probability to adapt TCP to the constellation is to distinguish between losses caused by congestion and by link errors. Explicit Congestion Notification (ECN) [RF99] is an example of this approach. In ECN, in case of losses caused by congestion, the sender will get a positive ECN feedback, while in case of losses caused by link reason, the sender will receive no ECN feedback. In this way, the sender can detect the congestion and lower the sending rate. However, sending ECN is expensive since it requires modifying all routers in wired terrestrial networks to get

a consistent standard of ECN notification.

Another probability is changing the sending rate recovery phase of TCP after congestion. UCLA's TCP Westwood [CGM⁺01] makes this choice. The sender of TCP Westwood just monitors the sending rate of its connection using ACKs inter-arrival times and estimates the bandwidth to see if congestion happens among the links to the destination. By using bandwidth estimation instead of using packet lossy feedback, TCP Westwood shows a significant improvement compared with ECN. However, a problem of these two approaches is that they only focus on the algorithm of congestion recovery. There is no better solution for the slow start phase.

TCP-Peach [AMP01] tries to address both the problems of “slow start” and congestion recovery. TCP-Peach replaces “slow start” and “fast recovery” of TCP with “sudden start” and “rapid recovery”. Both these two new phases use “dummy” segments with lower priority than real packets to explore the situation of the network. Because of the lower priority, the “dummy” packets will be the first dropped during congestion while they do not need to be recovered like the real data. When the sender receives the ACKs of “dummy” it will increase the sending window size with the exact number of “dummy” ACKs it receives. Thus the expansion of window of TCP-Peach is much faster than TCP normal. However, TCP-Peach is rather inefficient because the routers/satellites are always busy because they must process lots of segments including “dummy” that should be dropped.

3.4.3 TCP proxies

Beside the protocol modifications, architectural enhancement are also considered to improve TCP performance in satellite environment. This approach uses middleware technique and is generalized as Performance Enhancing Proxy (PEP) [BKG⁺01]. In PEP, the TCP connection is splitted into space/satellite segment and ground/nonsatellite segment. Gateways connect with each other via satellite links and connect end user via traditional TCP links. Gateways are mainly deployed on the ground, but recently gateway proxies are also considered onboard satellites [LSGS04].

TCP proxies communicate with each other via satellite links using private protocols which are transparent to the ground links. One approach is to use Space Communications Protocol Standards (SCPC) to implement congestion control scheme between two proxy gateways [SFCJ04], as shown in Figure 3.9. SCPC gateways notice the contention of transmission and control the sending rate, thus it provides congestion avoidance. Meanwhile, SCPC gateways not only provide congestion control inside satellite networks, but also take into account the congestions that happen on the terrestrial TCP links of the receiver side. When a loss is detected on the receiver side Internet, SCPC gateways on the receiver side will generate SNACKs, which means congestion happen on the other side of the path, and send SNACKs

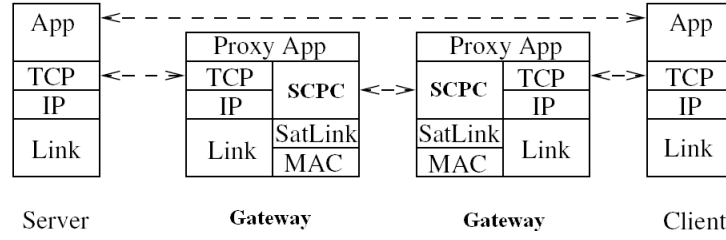


Figure 3.9: TCP gateways with SCPC

through the constellation back to SCPC gateways on the sender side and then reverse to the source. Thus the congestion control mechanism is implemented in the whole paths from the sources to the destinations.

3.5 Summary

In this chapter, we first discussed the intersatellite links in LEO constellation. Compared to bent-pipe non-ISL constellation, ISLs reduce data transmitting between ground and space segments thus saving radio resource usage and also reducing the data transmission delay. Then, we have outlined the protocol stack used in LEO constellations and analyzed some details of MAC layer and the transport layer. In chapter 4 we will focus on details of network layer of LEO networks and analyze related work that has been done recently.

Part II

Routing in LEO constellation

Chapter 4

Networking in LEO constellation

The high mobility of LEO satellites leads to a rapidly and regularly-changing network topology for the constellation, and raises a number of issues for the networking layer with respect to routing, particularly when considering adopting Internet protocols. In this chapter we discuss the problems of implementing routing in the satellite constellation. We will first discuss issues and problems of implementing IP directly in the constellation and then we propose the system model of LEO constellation for designing routing protocols and also related work of routing protocols.

4.1 IP over LEO constellation

In chapter 2, we have explained the existing routing problem in terrestrial Internet: instability, slow convergence and Scalability. However, from chapter 3, we know that LEO constellations have a high link error rate, high mobility and a longer propagation delay than the terrestrial links. The high link error rate causes the high probability of routes to change. The propagation delay time leads to a long convergence time. So if we directly extend terrestrial IP over satellites those Internet routing problems could be more severe. One possible solution is to modify the IP technique so that it could be more adaptive for LEO constellation. Wood has summarized these techniques that modify IP over LEO satellites [Woo01b]. IP tunneling establishes a virtual tunnel between two gateways through constellation. It routes packets of terrestrial Internet through the LEO network that belongs to a different routing realm and that can have a differing network layer. Packet formats, addressing space, and routing paradigms in the two networks may be entirely different. A virtual IP hop (a tunnel) is created between the two IP-capable gateways at the borders of the LEO network. When entering the tunnel, the terrestrial IP packet, including the IP header, is sent as payload data to the other side of the tunnel using the network layer of the LEO constellation. When leaving the LEO tunnels, the IP packets are reassembled as required and forwarded along the next terrestrial IP hop. Different from tunneling, IP NAT uses gateways to translate ad-

addresses between terrestrial IP and LEO satellites. NAT gains a space of address for a different separated routing realm. When entering the constellation, NAT translates the LEO realm addresses into new addresses in the address space that is suitable for use in the Internet realm for the IP packets. In this way, the terrestrial Internet can be prevented from noticing the topology changes of the LEO constellation. Anyway, in both of these two techniques, the LEO constellation is isolated from the Internet infrastructure as an extra autonomous system and implement private routing technique inside it. Different solutions are proposed to address the networking routing problem of LEO constellations.

4.2 Routing Concept and Topology Models

The routing problem in LEO constellation is usually formulated as optimizing value of a certain function that is represented by the value of edges in a graph. In most research work, LEO constellation is modeled as different types of graphs, two-dimensional graphs or three-dimensional graphs, that are composed of nodes connected by directional edges. Each edge has a value corresponding to the key function that is used to compute the route. In this way the routing problem is addressed so as to find an optimal path between a pair of nodes in a weighted graph.

To establish a system model is the most important issue that should be considered when people do research about networking in LEO constellations. Depending on the key word of the designed protocols, for instance to minimize the propagation delay or to maximize network throughput, different kinds of topologies are proposed as the system architecture of LEO constellations. Bellow are different types of architectures that have been used in recent years as LEO constellation models:

- **Pure Mesh:** In this model, LEO constellations are simplified as plain regular meshes that are composed by $M \times N$ satellite nodes. Nodes are assumed to connect with their neighbors by symmetric links. The weight of the links usually denote values corresponding to network loads, for instance the active transmissions or occupied bandwidth. This kind of model can be used to design algorithm that compute routes with minimum intermediate hop numbers, providing load balancing and also maximize network throughput.
- **Spherical Mesh:** This kind of architecture modeled a LEO constellation composed of M orbits each with N satellites as a three-dimensional graph. LEO orbits cross each other in north and south poles. On the different sides of the *Seam* plane the satellites move in different directions. The edges in this model contains more details about ISLs between satellites. ISLs are classified into two groups (*Interplane* and *Intraplane*). *Interplane* ISLs are shut down when satellites move into the polar zones. The weights of the edges denote the length or the propagation time of satellite links. This model is useful for

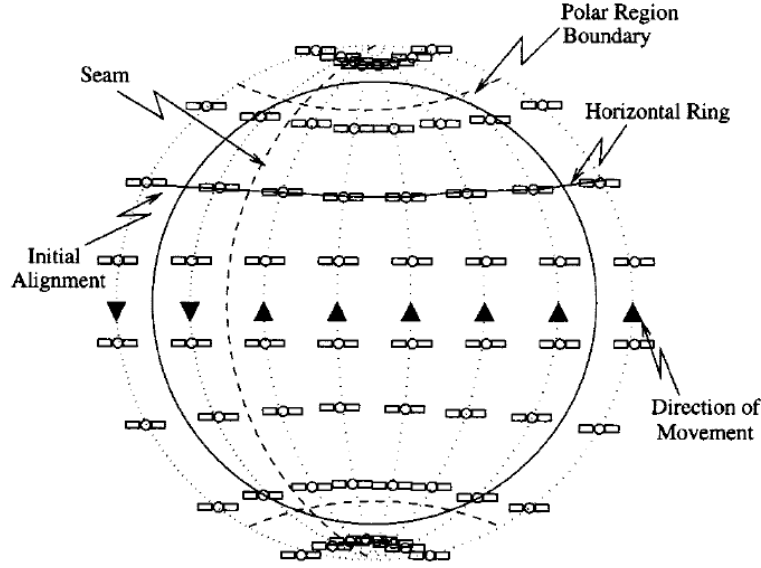


Figure 4.1: Spherical Mesh Architecture [EBA01]

routing algorithm that use optimal signal propagation time as the key function to compute routes.

- **SpiderWeb:** This model transforms the three-dimensional constellation into two piece of two-dimensional spiderweb topologies, one from the north pole side and the other from the south pole side[WYTW06]. Weights of edges denote also the length or propagation time of satellite links. Routing problems in this model can be equivalent to the problem of how to find the shortest path between two nodes in two planar spiderwebs.

4.3 Routing Classifications

Many routing algorithms in LEO constellation have been proposed in literature, each of them has their own emphases. Before we take a look at these related works, we will focus on the different key features of routing algorithms to see their properties, the main problems that they want to address and also maybe their limitations.

Figure 4.3 shows a routing classification of LEO constellation. But we should notice that a routing algorithm may belong to more than one feature of classification, for instance a routing algorithm can be online distributed and also about delays.

Here we will discuss some details about the classification:

- **Online vs. Offline:** Offline routing computes routes using the information that are pre-stored onboard each satellite. The information can not be changed

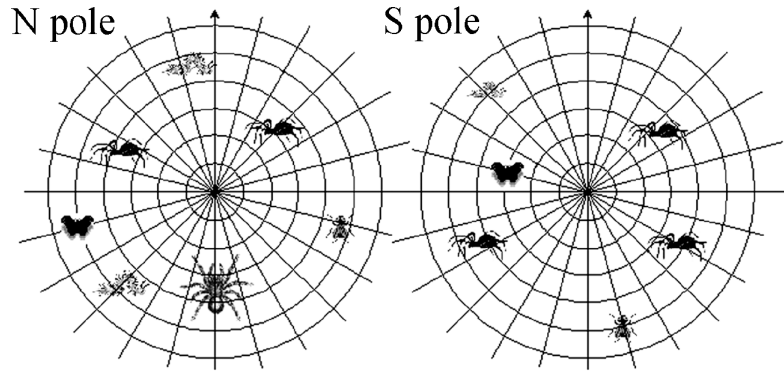


Figure 4.2: SpiderWeb Architecture [WYTW06]

thus the routing protocol using offline strategy is not flexible and can not provide fault tolerance for link and node failures. Online routing is more realtime and suitable than offline. But it may cause extra network loads;

- **Distributed vs. Centralized:** Centralized routing uses a network command unit (usually a terrestrial base station) to collect network information and compute routes. It reduces the computation onboard satellites but extra round trip time between satellites and ground station is needed for route computation and retrieving. Also the frequently changing topology of constellation leads to rapid restoration, which is a big disadvantage of centralized routing. Distributed routing needs more computation onboard each satellite but it is more adaptive for LEO's frequently topology changing. It is much easier than centralized routing to compute a recovery path after a failure.
- **Buffering vs. Non-buffering:** Buffer space is very important. It can be used to store existing routes. And it is also a buffer space for transferring data packets. Using buffering can reduce route computation because new packets can be forwarded following the routes in the buffer space. It also reduces packet loss in case of congestion because the packet can be temporarily stored in the buffer and then forwarded to follow new routes.
- **Single layer vs. Hierarchical:** Hierarchical satellite routing uses multiple layers of constellations together, for instance LEO and MEO. The MEO constellation takes responsibility to compute routing in the Hierarchical structure. Each MEO satellite manages the group of LEO satellites residing in its footprint. The LEO satellite just get routes from its parent MEO satellite and forwards the packets. Hierarchical routing can increase network throughput and reduce the computation of LEO satellites, but it costs more propagation delay of the round trip time between different layers of satellites;

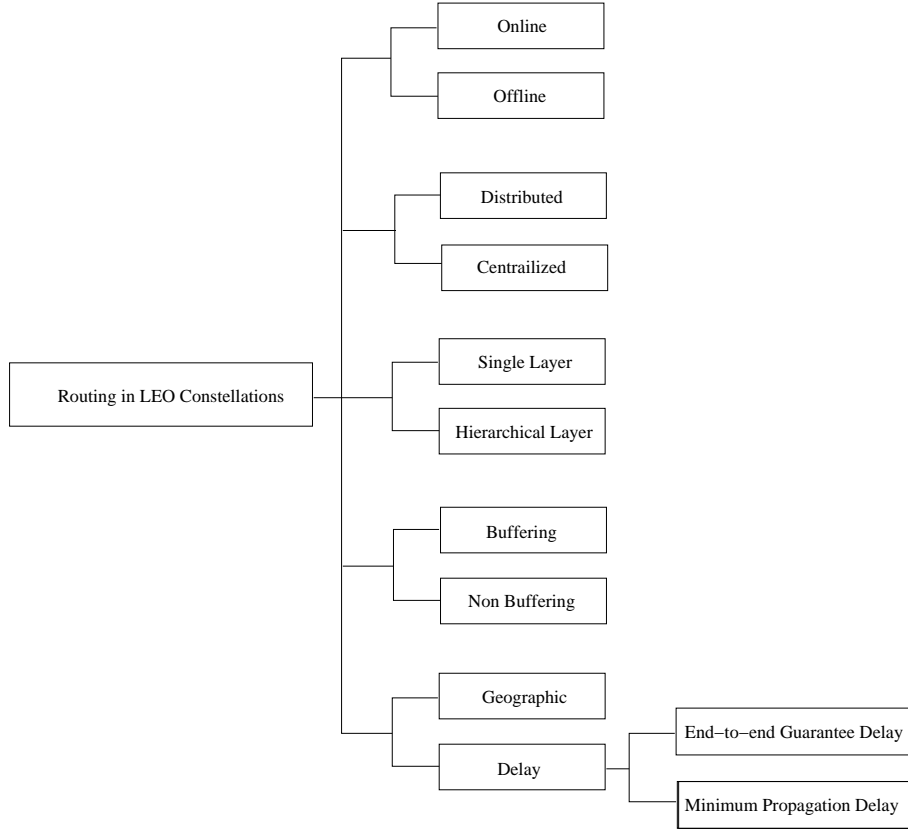


Figure 4.3: Routing Algorithms Classifications

- **Geographic vs. Delay:** Actually, geographic and delay are the two objective functions for routing. We put them here as one of the classifications because they are prevalent in routing in LEO constellation. Geographic routing usually takes into account the hop number of routes or whether the path is the shortest one, while delay routing is more concerned about the propagation delay between satellites. And even delay routing can be classified into two subsets. One is to compute a route with minimum end to end delay time, the other is to compute a route using the higher bound of delay guarantee of services.

4.4 Related Work

In this section, we will discuss the routing algorithms which have been proposed in the literature, following the routing classification given in Section 4.3.

4.4.1 DT-DVTR and other offline routing

As we mentioned in Chapter 2, offline routing has been thought of as a candidate for routing algorithms for ISL routing. Discrete-Time Dynamic Virtual Topology Routing (DT-DVTR) [Wer97, HL01] is one candidate for offline routing algorithms. DT-DVTR makes full use of the periodic nature of satellite constellation and works completely offline. It divides the system period into a set of time intervals so that the topology changes only at the beginning of each time interval and remains constant until the next time interval. In each interval, the routing problem is a static topology routing problem. The constellation is modeled as a mesh on which shortest path algorithm is implemented to compute the routes. Because of offline computation, we need a large storage on board for routing table which will compensate for reduced on-line computation complexity. Even if an optimization procedure can be used to choose the best path (or a small set of paths) from the series of instantaneous routes to reduce the storage size, some links may become congested while others are underutilized.

Another detailed offline algorithm is proposed in [RE00]. This algorithm is offline and distributed. A two-dimensional graph is used as the system architecture. The whole system time is divided into k snapshots, each of which is denoted by a graph G . Each link of G is weighted with a cost. Once data packets are generated at the source node, a shortest path algorithm is applied on top of the link matrix to find the best path from the source to the destination node. The cost of the link in the graph is computed by the propagation delay of the link between satellites and two preferences. One preference corresponds to the link stability, if the link is a permanent link (intraplain ISL) then it has higher preferences than a temp link (interplain ISL). Another preference denotes the value of the link in the global traffic demand pattern. Usually, the link residing in a hot spot area has a low preference. Moreover, handover is also taken into account in this algorithm. To represent handover, another cost is given to each link to denote the duration of the link. The best path is then chosen by the cost of link and the cost of duration together.

This offline algorithm is simple also in order to try to provide load balancing by using the preference of traffic pattern. But since it is completely offline, it is not flexible and adaptive for a constellation realm which has high link error rates. Once a link or node error happens, those routes containing the failure are not available, and data loss occurs until the next satellite comes.

4.4.2 Predictive Routing

In [EKDT00], a centralized algorithm, Predictive Routing Protocol (PRP), is proposed. This protocol implements centralized routing at ground gateways. It is specialized for CBR and VBR traffic. The routes are computed using the residual bandwidth of ISLs. The satellite network is represented by a directional graph G ,

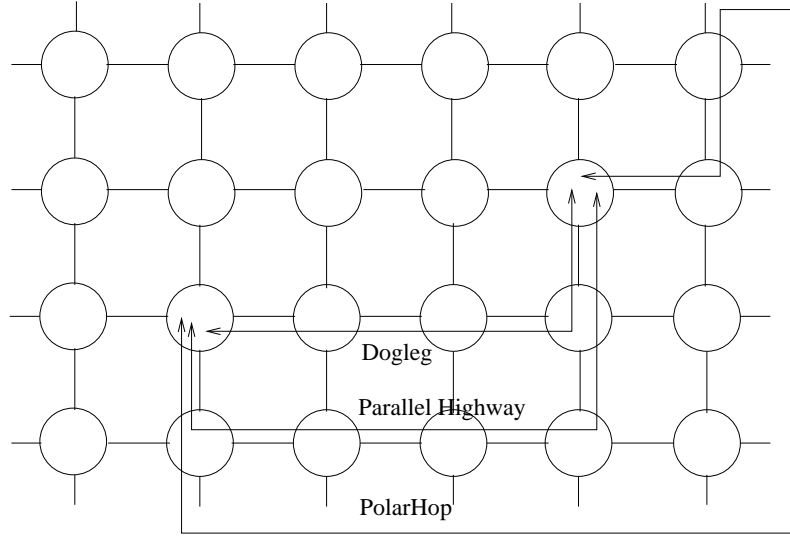


Figure 4.4: Dogleg; Parallel Highways and Polar Hop algorithms

with a vertex representing a satellite node, and an edge representing an ISL. Every edge has a weight which is the residual bandwidth available on the ISL. This protocol tries to distribute the total traffic in the network and minimize the traffic on the most congested ISLs. It has good behavior on load balancing, but the routes it computes may not follow the minimum hop path and may have a longer delay. And as a centralized routing protocol, it has a big disadvantage for route recovery after failures. Once link or satellite node failures happen, communication between control unit gateways and the satellite it is necessary to recover new routes.

4.4.3 Dogleg, Parallel highways and Polar hop routing

Paper [KE03, MR97] introduces three very simple distributed routing algorithms for LEO satellites: *dogleg*, *parallel highways* and *polar hop*.

The Dogleg algorithm directs a packet towards the destination satellite firstly by horizontal links and secondly by vertical links, or the converse, but never with zig-zag paths.

The Parallel Highways algorithm distributes the load among three routes that are parallel to the one which has the minimum number of hops (that used by Dogleg algorithm, for instance). In this way, the load is spread over many links, so as to avoid bottlenecks in those satellites that are overloaded. All packets belonging to the same source-destination pair are assigned to the three paths randomly, and the path to which they belong is written in the packet header to simplify the switching

at each satellite they cross.

The Polar Hop algorithm, instead, takes advantage of the fact that on polar regions, the satellites receive a small number of requests coming from ground users. So they are good candidates to serve as store-and-forward satellites. The algorithm dictates routes that always cross the links on the polar regions, even if the packets have to go from East to West, or vice versa.

All above three algorithms route packets using only geographic positions, so we can expect that they either easily lead to congestion or have high propagation delays. Dogeleg always routes traffic via the same shortest paths from the sources to the destinations. Thus congestions can occur very often near hot spot areas. Parallel Highways try to avoid using the shortest paths so that congestion may be reduced but this algorithm is not flexible, all the traffic are routed using the same strategy, and congestions may still happen. Polar Hop always routes the traffic via two polar hops. Definitely this will balance the traffic load to reduce congestion but the problem is that all the traffic has very long propagation delays which will greatly influence the Quality of the Service. Results of simulations about these three algorithm will be provided later in Chapter 5. However, the ideal routing protocol to be used in LEO constellation should have a short propagation delay and also provide congestion avoidance.

4.4.4 Greedy routing using buffer information

To provide real time load balancing recently Greedy routing has been used. A new routing scheme called Explicit Load Balancing (ELB) is proposed in [TJKN05]. In ELB, satellites periodically send greedy information to their neighbors to notify them of the status of their buffers. Congestion can be measured as buffer “load”. ELB sets two thresholds for the queue size and classifies the buffer into three status: Free, Fairly Busy, Busy. Usually, the “Busy” status means that congestion has occurred. And ELB builds the routing cost metrics to find the shortest path. Combined with the greedy information of the next hop, once the next hop of the shortest route gets congested, ELB will compute another alternative path. By this way, ELB provides congestion avoidance and achieves load balancing.

4.4.5 Minimum propagation delay routing

For LEO satellites, propagation delay is one of the most important critical issues that should be taken into account when designing a routing algorithm for the constellation network.

While using propagation delay as the key factor for algorithm, first of all, the constellation architecture and system running model should be carefully established.

Spherical mesh topology could be a good candidate here.

A secondary problem for propagation delay routing algorithms is how to provide a congestion control mechanism. Normally, the packets will be forwarded following the shortest path via propagation delay or the ISLs length from the source to the destination. In this case, if the satellite is a hop spot area, then congestion may happen in a certain direction of ISL with high traffic requirement.

Investigation about minimum propagation delay routing was proposed in [HK00]. In this paper, minimum propagation delay protocol is compared with the shortest path algorithm using geographic information. Two packets were routed between a pair of randomly global chosen points by the simulation. One packet uses a global shortest path algorithm based on minimizations of the propagation delay, the other was routed via the distributed geographic-based shortest path algorithm. The performance results discussed in that paper show that generally the routing strategy that uses minimum propagation delay has better behavior than using geographic-based algorithm. However, the experiment in this paper does not show the influence of link errors and node failures. Neither is congestion taken into account in this paper.

In [EBA01], a datagram routing algorithm was proposed based on a polar orbits LEO constellation model. In this paper, the transmission delay between a pair of adjacent LEO satellites via an ISL is called *individual propagation delay*. The sum of the *individual propagation delays* of a multi hop path is called the *total propagation delay* of this route. Minimum propagation delay protocol just picks up the route with a minimum *total propagation delay* among the multi hops routes set from the source to the destination. Now we will discuss an approach proposed using minimum propagation delay.

The minimum propagation delay route is calculated regarding the latitude positions of the source and destination satellites: 1) if the source and destination satellites are at different latitudes, source latitude is higher than the destination, for instance, (shown in figure 4.5), then the minimum propagation delay path passes through the satellites in the same horizontal ring as the source satellite (S_0 in figure 4.5) and is in the same orbit plane as the destination satellite (S_n in figure 4.5); 2) if the source and destination satellites are at the same latitude, then the minimum propagation delay path involves all horizontal hops on any horizontal rings between the ring of source/destination and the polar region (shown in figure 4.6).

Datagram routing algorithm first determines which satellite should be the next hop in the minimum propagation delay. Then it uses onboard packet queue length to the next hop in the buffer to provide congestion avoidance. The advantage of this algorithm is that in this way, no extra traffic load information will be exchanged. The drawback is that only congestion of one hop further is considered in the whole route. Long end to end delay still may happen in some high traffic area.

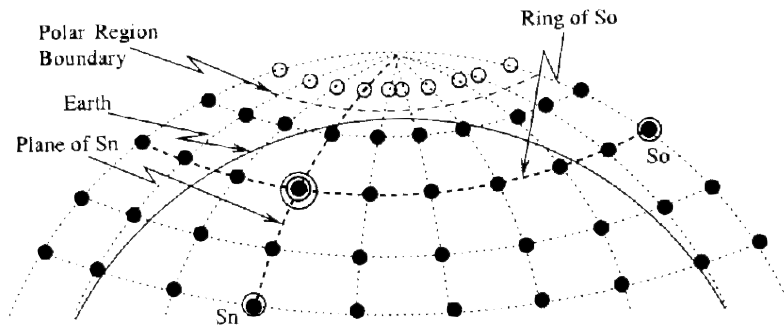


Figure 4.5: Datagram routing: nodes in different latitude

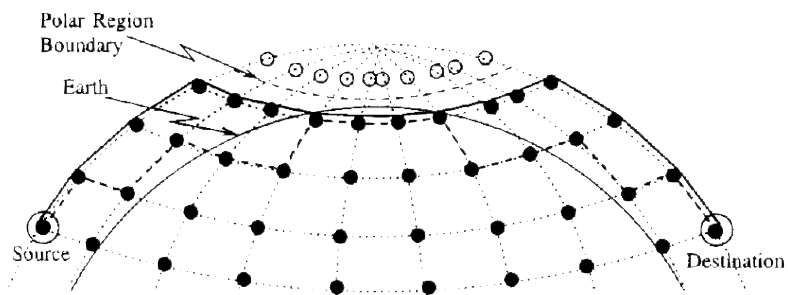


Figure 4.6: Datagram routing: nodes in same latitude

Minimum propagation delay protocol is able to the nature to support QoS and it works without any network overheads. However, congestion avoidance still can not be overcome by this technique. Extra notifications are still needed to specify the link errors and congestions.

4.4.6 End-to-End Delay routing

The minimum propagation delay routing mentioned above is a best effect choice method. Another kind of routing scheme is also based on the propagation delay, but using the delay bound requirement to provide end-to-end delay guarantee [HYK04, HYK05].

A new routing algorithm called Satellite Routing for End-to-end Delay (SRED) is introduced in [HYK04]. The authors assume that there is an upper bound D of end-to-end delay predefined by each classes of service requests. SRED introduces Weighted Fair Queuing (WFQ) [WC96] to provide the delay guarantee. Following the inverse formulation of WFQ, the routing algorithm can decide how much bandwidth should be allocated to a connection given a certain upper delay bound and certain traffic characteristics of the connection. SRED has high performance for multimedia request in LEO networks. However, SRED only considers connections between a pair of individual nodes. Interference between global pairs of nodes are taken into account later when High Performance Satellite Routing (HPSR) algorithm is designed [HYK05]. Similar to SRED, HPSR also allocates identical bandwidth to all the links of a connection. But different from SRED, the computed bandwidth requirement of HPSR is lower than the average transmission rate of a connection. This is because HPSR puts the service requests of all nodes together into a global pattern to build connection service curves. These curves are in increasing order so that a minimum bandwidth that satisfied the upper bound of end-to-end delay for the service can be computed.

The end-to-end delay guarantee routing scheme provides good performance when supporting multimedia services. But for the best effect requests which may not have a predefined delay bound, this approach is not suitable in this case.

4.4.7 Hierarchical group routing and cluster routing

Multiple layers of constellations are considered for routing protocols. People are thinking of dividing LEO constellation into groups according to the footprints of MEO constellation. MEO satellites are used to reduce the overheads for the network thus providing better performance. Some proposals will be discussed here about using LEO/MEO together, and a single LEO layer cluster routing algorithm

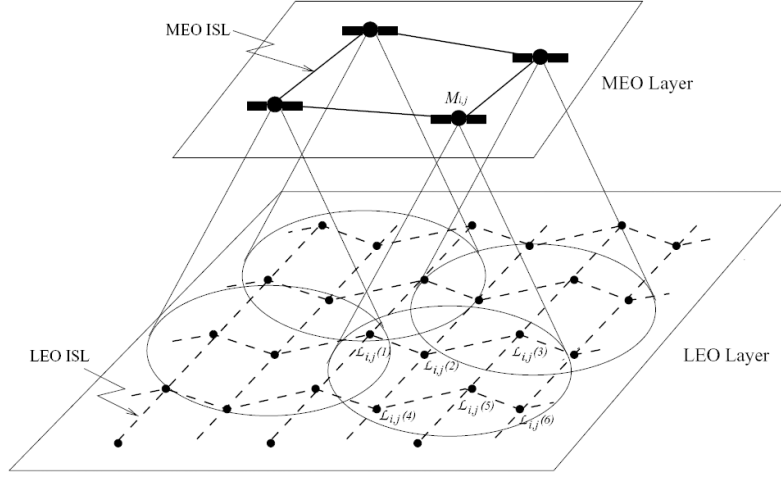


Figure 4.7: LEO/MEO hierarchical architecture

will be examined as a comparison.

Hierarchical architecture of satellite routing concept is proposed in [LLKK00]. In this paper, the multilayer model is established by using cluster. Satellites in the lower layer are clustered to form satellites in upper layer.

Minimum propagation delay routing is the basis for a new Satellite Grouping and Routing Protocol (SGRP) proposed in [CE05]. In this protocol LEO satellites are divided into groups according to the footprint area of the Medium Earth Orbit (MEO) satellites. The hierarchical LEO/MEO architecture is shown in figure 4.7. Collaboration between LEO and MEO satellite layers are utilized in SGRP: MEO satellites compute the routing tables for the LEO layer. The main idea of SGRP is to transmit packets in minimum-delay paths and distribute the routing table calculation for the LEO satellites to multiple MEO satellites. LEO satellites are divided into groups according to the footprint areas of the MEO satellites in each snapshot period. Snapshot periods are determined according to the predictable MEO trajectory and the changes in the LEO group memberships. The MEO satellite that covers a set of LEO satellites becomes the manager of that LEO group. Group managers are in charge of collecting and exchanging the link delay information of the LEO layer, and calculating the routing tables for their LEO members. LEO satellites receive routing tables from their group managers.

Using SGRP, the calculation of the routing tables is shifted to MEO satellites, which effectively distributes the power and onboard resource consumption of LEO satellites. But more problems between LEO and MEO constellation, like roaming of LEO satellites among different MEO footprints, propagation delay between

LEO/MEO and MEO themselves, should be considered.

To supply SGRP, a routing scheme between LEO and MEO constellations is proposed in [YZL05]. This work only considers the communication between LEO and MEO satellites. The links between LEO and MEO satellites depend on the visibility of each other. The link costs in this work consider either the minimum link distance or the maximum link life time. However, only simple OSPF algorithm, which is not suitable to the satellite environment has been tested. More research should be done taking into account the intermix routing of intra and inter LEO/MEO constellations.

Actually another group routing algorithm is proposed before SGRP in [LMYW04]. This approach also divides the LEO constellation into clusters, but instead of using MEO satellite as the group manager in each cluster, one of the LEO satellites in the cluster is chosen as the manager node. Normal nodes in a cluster only know about the link state inside the cluster. Link status only spreads out among cluster manager nodes. Then, a border satellite based source routing algorithm generates routes to forward the packets. Link failures are also considered in this algorithm. But are the relations between network overheads and the performance of the algorithms, for instance how the link states are broadcasted among the constellation, periodical or just dependent on the link failure? Moreover, none of the above approaches has taken into account the real Internet traffic patterns and classifications when designing their own protocols.

4.4.8 Routing using Markov Decision Process

The discussion of SRED and HPSR shows that these two algorithms support multimedia service well. But there is another solution that applies Markov Decision Process (MDP) [Kri90]. A new algorithm, Adaptive Cost Routing (ACR), is proposed in [TKW04]. By applying MDP formulation using the parameters of service bandwidth requirement, the total link capacity, the arrival rate and its holding time, a reward value will be given to a link once a new connection is established. In other words, the admission of a media request gains a reward but pays by consuming a certain amount of bandwidth resource. ACR picks up a route with the sum of the cost over all links being less than the predefined reward value of the incoming request.

ACR can support multimedia services with good bandwidth allocation. But ACR uses a checksum of the link cost to choose a route which may cause congestion in portion for the route.

4.4.9 Traffic class routing

To accommodate a variety of applications with diverse performance requirements in next generation of satellite networks, adaptive routing procedures supporting different levels of services are required.

In [MWSK02], the authors implemented Dijkstra shortest path algorithm on top of a link cost metric that is composed of both traffic load and the propagation delay on the links. The algorithm is evaluated in both uniform ISLs traffic load and a more realistic traffic scenario of globally homogeneous source/destination distribution. The results in this paper reveal that the coefficients chosen for traffic load and the propagation delay in link cost metrics to compute routes can influence the behavior of average packet delay.

Moreover, the literature [SMK⁺04] analyzes in detail the internet traffic scenarios with different issues like, packet interarrival time distribution, packet length distribution, etc. then introduces a new TRAFFIC CLASS DEPENDENT (TCD) algorithm. In the paper, the real internet traffic flow is then classified into three different classes:

- Traffic class A: typical applications belonging to this traffic class include interactive real-time applications, such as voice-over-IP (VoIP) and interactive video applications, which require the delay to be minimized.
- Traffic class B: representative applications of this traffic class are video-on-demand (VoD) and large file distribution, which require the throughput to be maximized.
- Traffic class C: this traffic class represents best-effort service as known in Internet and is meant for applications without any specific requirements. This traffic class is expected to be available at a low price in exchange for the reduced quality-of-service (QoS).

The TCD algorithm computes routes completely depending on the traffic class definition, it uses different parameters to compute routes for different traffic classes. Link-cost metrics for the traffic classes A and C are typical additive metrics with the values of the propagation delay plus onboard queue delay. Thus, the shortest routes are calculated over the link-cost metrics using the Dijkstra algorithm. On the other hand, the link cost for the traffic class B is a concave metric with the values of available bandwidth onboard each satellite. The route computed from the concave metrics of class B finds the paths within minimum hop count with the maximum available bandwidth.

The TCD algorithm is adaptive to the real Internet scenario. It can provide different QoS to different traffic requirements. It chooses routes with maximum available bandwidth for high QoS traffic class B and computes minimum propagation delay paths for class C with non QoS requires. The criterion is that many parameters chosen for the route computation of different traffic class increases the complexity of the algorithm. Also the onboard network load should be exchanged to support the route calculation of class B.

4.5 Routing algorithms summary

In this chapter, we have first outlined different types of system models for LEO constellations.

Secondly, we have given routing classifications based on the different characteristics of the routing protocols and the emphases of the problem that is being addressed.

Finally, we have analyzed different routing strategies proposed in the literature. Offline routing is simple but not flexible for load balancing and failure tolerance. Geographic routing is very simple but static and easy to lead congestion. Minimum propagation delay routing computes the routes without any traffic load but need to consider congestion avoidance. End-to-end delay guarantee routing can provide good performance for multimedia service but is not good for best effort routing. Group routing mainly needs to globally exchange load information. Finally, the traffic class dependent routing is quite expensive and also needs control information exchanging.

As we mentioned before, the ideal routing protocol for LEO constellation should be both fast and with congestion avoidance. In the next chapters, we will demonstrate our protocol which is very simple, fast and has special congestion control mechanisms.

Chapter 5

Our Control Route Transmission (CRT) Protocol

5.1 Motivations

As we mentioned earlier Chapter 2, Internet provides services for packets with TCP heads over IP. In traditional terrestrial IP, the network does not normally enforce traffic shaping of flows or allocate capacity used for flows. If packets are lost due to congestion there is no explicit feedback from the network to the source to provide notification that congestion has taken place – since the network is congested, there is an argument against generating even more notification traffic to add to the congestion.

TCP implements its own end-to-end feedback, via acknowledgements, to ensure a stable channel and reliable delivery. The performance of standard TCP can be improved by increasing the initial window [AFP02] or other enhancements [AGS99, AMP01]. However, the core reaction of TCP/IP is that, the TCP sender notices the missing of TCP segments or acknowledgements and assumes that the cause of this is congestion in the network, and will then exponentially decrease or back off the rate of sending new data segments. This assumption can lead to loss of throughput on less-reliable links such as error-prone satellite links, where packets are lost due to errors in the channel on the satellite link, rather than due to congestion at terminals [Woo01a].

Differently from TCP, in which the minimal information available on the state of the internetwork via feedback is provided by acknowledgements, UDP packets are one-way and delivery is unreliable. Moreover UDP does not use end-to-end feedback to alter its transmission rate. This means that UDP-based applications, including multicast applications, must implement their own congestion avoidance routines and ensure that they receive data with any necessary degree of reliability inside the LEO

constellation [Woo01a].

To summarize the traditional Internet protocols over IP, TCP and UDP, they all have different drawbacks that can not be directly applied to LEO constellation. TCP relies on acknowledgements and can not distinguish congestions and link errors. Also TCP's slow recovering transmission window can not efficiently use the ISLs' bandwidth after congestions. On the other hand, UDP protocol has implementing difficulties like bandwidth reservation and congestion avoidance in LEO constellations. The aim of our work is to design a new network protocol with good behavior that can carry terrestrial IP traffics through LEO constellation. We assume that TCP and UDP traffics are encapsulated into the packet format that our protocol uses before entering the constellation through the terrestrial base stations. The packets will then be unpacked by the base station on the destination side. In this way, we can avoid all the problems caused by the TCP and UDP protocols.

In Chapter 4, we have discussed several routing algorithms internal to LEO constellation. But as far as we know, they either do not completely provide congestion avoidance or take into account congestion but still present disadvantages. Offline routing algorithms like DT-DVTR [Wer97, HL01] use pre-stored information to compute routes can not model congestion on satellites and ISLs. In this protocol, data loss may happen due to packets continuously being forwarded to congested areas. On the other hand, the online routing algorithms proposed so far present many disadvantages even if they take into account congestions. Dogleg [KE03][MR97] algorithm uses the shortest path route to deliver messages without considering congestion. Parallel Highway and Polar Hop algorithms [KE03][MR97] use fixed rules to forward packets to avoid hot spot areas, but they are not adaptive. If congestion arises they can not adjust the route to avoid hot spots. Datagram algorithm [EBA01] uses minimum propagation delay to compute routes and uses the onboard buffer information to provide limited congestion avoidance and can not prevent the packet from being forwarded to hot spot areas. Greedy routing [TJKN05] has limitations similar to Datagram algorithm. It can only avoid the congestion of the ISL that connects the current satellite to the next one. It does not take into account congestions in all the ISLs in a route. End-to-end delay routing [HYK04, HYK05] does not focus on congestions but only the delay bound of the data request. The algorithms above are not satisfactory for congestion avoidance. In this thesis, we want to take a step forward and design a routing algorithm that is adaptive to global traffic conditions so as to provide real time congestion avoidance and global traffic balancing.

Our goal is to design a routing protocol which provides traffic balancing to increase network throughput. Our routing algorithm is specifically designed for stream burst traffic. The basic thinking of our protocol is to use control messages containing congestion information as an explicit notification of link usage. Using these periodi-

cal control messages, a global traffic matrix is maintained to hold traffic information and to compute routes. Our algorithm divides system time into small time epoches. The control messages are updated at the beginning of each time epoch and globally exchanged. Thus the routes are computed with realtime congestion avoidance. Furthermore, the control messages can also contain bandwidth reservation parameters to provide QoS requirements. (This extension will be discussed in Chapter 6.)

5.2 Constellation model and traffic model

5.2.1 Constellation Architecture

We need constellation addresses of satellites to be able to send a packet through a ISL from one satellite to another. The constellation addresses could be pre-configured based on the *Virtual Node* (VN) concept [Woo01b, HL01, MR97]. The VN concept tries to hide the mobility of satellites to the routing protocols. A virtual topology is set up with VNs superimposed on the physical topology of the satellite constellation. Even as satellites are moving across the sky, the virtual topology remains unchanged. States such as routing table entries and channel allocation information are transferred continuously from one satellite to another. Routing is performed in the fixed virtual network, by using a common routing protocol. VNs are known by all entities in the constellation AS including ground base stations and satellites. Once a terrestrial IP router wants to communicate with another IP host on the other side of the constellation AS, the base station that has detailed knowledge of both the satellite constellation and the terrestrial Internet will treat the request with Network Address Translation (NAT), as shown in figure 5.1 [Woo01b]. Thus the constellation is isolated from the Internet as a private network and the information needed to implement routing in constellation is greatly reduced, only constellation address information is used to compute routes, for instance:

- Satellite or VN IDs.
- ISLs interfaces to forward data.

Beside the satellite addresses, ISLs are another important object in the constellation architecture. We assume that all the ISLs in the constellation have the same bandwidth. Each satellite has four outgoing ISLs and four incoming ISLs that connect with its four neighbors. The same structure is also used in the [SM04] as a model for LEO satellite networks. ISLs use FDMA/TDMA (Chapter 3) to support multiple connections between neighboring satellites.

5.2.2 Our Constellation Model

As we have discussed in Chapter 4, to solve the routing problem, LEO constellation is usually modeled as a weighted directional graph. Weights of the edges in the

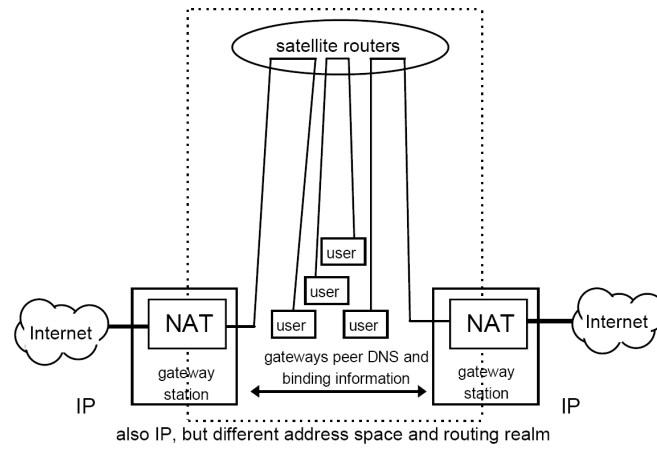
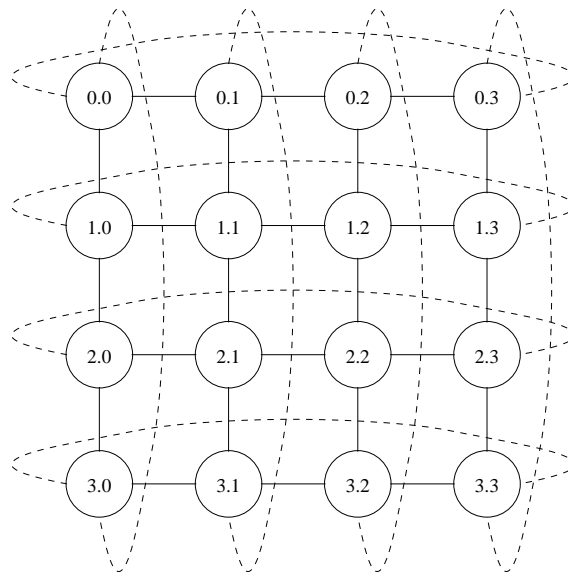


Figure 5.1: NAT between Internet and constellation [Woo01b]

Figure 5.2: 4×4 mesh

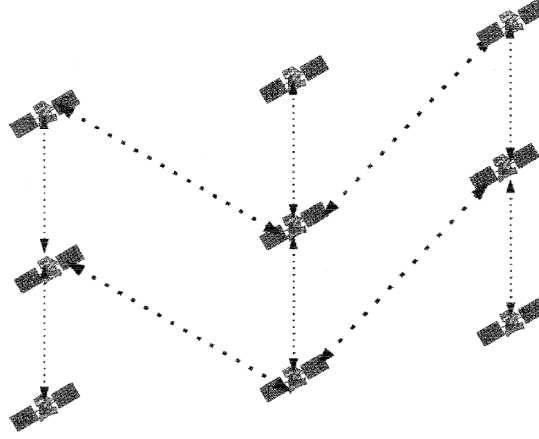


Figure 5.3: A satellite and its immediate four intersatellite links in Iridium [Gav97].

graph are used in the objective function to compute routes. We also model the constellation as a graph, where each node denotes a satellite and each directed edge denotes an ISL. In our model nodes are interconnected as a two-dimensional mesh and edge weights correspond to the number of *active transmissions* on the edge (Figure 5.2). As we said, our algorithm focuses on stream bursts. We assume that terrestrial gateways generate requests for bursts containing a certain amount of packets. When a satellite receives a request, it computes a route, establishes a connection and then transmits the burst. We call *active transmission* the number of bursts being transmitted on the ISL corresponding to an edge at a given time.

In our model, we assume that each satellite has exactly four ISLs to communicate with its neighbors in the same or neighboring orbit (figure 5.3 shows the satellite in Iridium constellation with ISLs [Gav97]). We also assume that there are symmetric edges in both directions between a pair of neighboring satellites in the mesh topology. Each directional edge is weighted with the number of active transmissions on the link. The objective function we use to compute routes is the minimization of the sum of the weight of edges in the route.

5.3 Control Route Transmission (CRT) Protocol

CRT schedules message bursts of variable lengths from a source satellite to a destination satellite. The main idea of CRT is using periodically exchanged congestion control messages to build a congestion matrix then to compute the shortest path on this matrix and pick up the best route for all pairs sources/destinations. CRT was first proposed in [HP06].

Table 5.1: Stream traffic structure

START	packet	...	packet	STOP
--------------	--------	-----	--------	-------------

5.3.1 Stream format

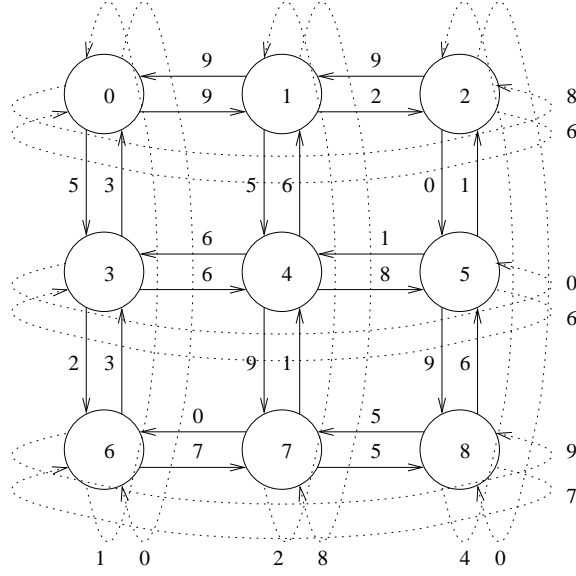
We assume that the transmission of a burst from earth is required to the source satellite with a header specifying the message destination which is encapsulated by the terrestrial base station. The satellite acknowledges this request accepting or denying transmission. These stream bursts have beginning heads and end notifications. We name them **START** and **STOP** segments. These two segments are very important in CRT protocol. In the basic version for our protocol, both the **START** and **STOP** are only notification segments used to update the network loads which will be used to generate the control messages (Table 5.1). When a transmission is accepted, the source satellite computes the best route for this message to reach its destination, generates a **START** packet for the burst containing all the routing information, and sends it along the chosen route followed by all the subsequent packets in the burst. When the burst is finished the source satellite generates a **STOP** packet, which is sent along the same route and signal to all the intermediate nodes at the end of the burst.

5.3.2 Computing the best route

CRT works on time epochs of length Δt (let's say for instance, 0.05s). At the beginning of an epoch each satellite broadcasts a control message (CM) to all the others. A CM message says how many bursts are currently being transmitted on the outgoing ISL of its source satellite.

The CM messages received by a satellite are used to build a directed graph G which records the current congestion status of the constellation. A node in G represents a satellite and an arc (i, j) corresponds to an ISL connecting satellite i with satellite j . Arcs are weighted, and $w(i, j)$ records an approximation of the number of bursts currently being transmitted along ISL (i, j) . Initially all weights are zero. When satellite s receives a CM from satellite k , it updates G , computes the new shortest paths from s to all the other satellites, and records them in a Shortest Path Table (SPT).

Figure 5.4, shows a possible G graph for a 3×3 mesh, weights on the arcs represent outgoing bursts of traffic. On a satellite, CM messages are recorded in a congestion matrix C , with a row for each satellite and a column for each direction of outgoing link. Table 5.2 shows the congestion matrix describing the situation depicted in Figure 5.4.

Figure 5.4: 3×3 Mesh With Traffic Information

When a new CM message arrives the congestion matrix is updated. At the beginning of each time epoch the congestion matrix is used to build an adjacency matrix for G (Table 5.3). The adjacency matrix is used to compute the shortest path from satellite s to all the others (results in Table 5.4). The shortest paths are recorded in SPT and will be used to route messages until the current epoch ends.

Table 5.2: Congestion matrix

Sat ID	North	West	South	East
0	0	8	5	9
1	8	9	5	2
2	0	9	0	6
3	3	0	2	6
4	6	6	9	8
5	1	1	9	6
6	3	9	1	7
7	1	0	2	5
8	6	5	4	7

5.3.3 Routing a new burst

Consider a request for a new burst b from source s to destination d . The source satellite s first computes a *unique identifier* for b ($uid(b)$). Then it accesses SPT to find out which is currently the best route $route_b$ to reach d from s ($route_b = r_1 = s, r_2, \dots, r_k = d$).

Finally, it creates a START packet for b in which it inserts $uid(b)$ and $route_b$ and sends it out towards the first node in the route (r_1). All the subsequent data in the burst will be inserted into fixed length packets of two fields:

- $uid(b)$
- data

and delivered along the same route. At the end of the burst a STOP packet will be generated and sent along the same route to signal the end of the burst.

A generic node r_i in $route_b$ will receive the START packet, insert the pair ($uid(b), r_{i+1}$) in the routing table and route all the subsequent packets with identifier $uid(b)$ to r_{i+1} . When the STOP packet for $uid(b)$ is received the corresponding entry in the routing table is freed.

The destination node will simply recognize from the START packet that it is the final destination of the burst and remove the packets from the constellation taking care of their delivery to the final destination on earth.

START/STOP packets are also used to update the congestion information on each satellite. In particular, the receipt of a START packet in satellite r_i will increment the number of active transmissions on the relevant incoming outgoing ISL on row r_1 of the congestion matrix and the receipt of a STOP packet will do the reverse decrement.

5.3.4 A simple example

For instance, assume that there is a new burst from satellite 0 to satellite 5 in Figure 5.4. From the final all-pairs shortest paths table (Table 5.4), the best path from 0 to 5 is (0,3,5). Thus the burst is processed as follows. Node 0 generates a unique identifier (say $uid = 98998900$, the last two digits are the unique index of source satellite) for the burst and creates a START packet containing (98998900,(3,5)) and propagates it to node 3. Then, C is updated locally incrementing $C[0][south]$ by 1 and pair (98998900,3) is inserted into the routing table. When the start packet is received, satellites 3 and 5 insert the pair (98998900,5) and (98998900,END) in their

Table 5.3: Adjacency matrix

0	9	8	5	∞	∞	0	∞	∞
9	0	2	∞	5	∞	∞	8	∞
6	9	0	∞	∞	0	∞	∞	0
3	∞	∞	0	6	0	2	∞	∞
∞	6	∞	6	0	8	∞	9	∞
∞	∞	1	6	1	0	∞	∞	9
1	∞	∞	3	∞	∞	0	7	9
∞	2	∞	∞	1	∞	0	0	5
∞	∞	4	∞	∞	6	7	5	0

Table 5.4: Shortest paths

0	9	4	3	4	3	0	7	4
8	0	2	8	3	2	7	7	2
6	7	0	6	1	0	5	5	0
3	7	1	0	1	0	2	6	1
9	6	7	6	0	6	8	9	7
7	7	1	6	1	0	6	6	1
1	9	4	3	4	3	0	7	4
1	2	4	3	1	3	0	0	4
6	7	4	8	5	4	5	5	0

respective routing tables. Satellite 3 also updates the congestion matrix incrementing $C[3][\text{west}]$ by 1.

When a data packet is received with uid=98998900, all nodes use the routing table to process it to its destination. At the end of burst, the source satellite creates a STOP packet with uid=98998900 and deletes the corresponding entry from the routing table. Also $C[0][\text{south}]$ is decremented. When a stop packet is received, satellites 3 and 5, update their routing tables and 3 also decrements $C[3][\text{west}]$ by 1.

5.4 Performance evaluation

We evaluated the performance of CRT using three main tests: measurements of the packets dropping rate, measurements of the average end-to-end delay time and measurements related to different lengths of update epoches. In all tests we compared CRT with other algorithms in the literature whose behavior is comparable

with CRT. As we said, the goal of our algorithm is to provide congestion avoidance between the entire routes and also balance the global traffic. If our algorithm shows a smaller packet dropping rate than other algorithms under the same simulation environment, it means that less packets are dropped by congestion or less congestion happens. In this case, our algorithm achieves its main goal. On the other hand, our algorithm may use a route which is not the shortest one to transmit a packet. We should know if the end-to-end delay of our algorithm is comparable with other algorithms. In this case, the testing of average end-to-end delay measurement can help us.

Finally, our algorithm works in time epoches. The frequency of global control message exchanging has an influence on the performance of the algorithm. Higher network overheads may happen due to very frequent control message updating. On the other hand, congestion may occur if the update frequency is too low. We can understand the relationship between the performance and time epoches by choosing different epoch values. In the following section we discuss how we set up our simulations experiment and will also discuss the results.

5.4.1 Simulation tools

There are lots of simulators that can be used to evaluate networking layer algorithms. We will discuss some popular and special simulation frameworks in this section and give motivations for our final choice.

NS-2

NS2 [UP06] is a discrete event network simulator developed at the University of California at Berkley (UCB). It began as a variant of the REAL network simulator in 1989 and has eventually evolved. NS2 takes full advantage of the features of object-oriented programming. It is written in C++ and OTcl. Though it does not guarantee production of a faithful replica of the real world, it does try to model accurately most of the protocol behaviors and can be used to study various protocols at different levels of the OSI layers. It is focused on modeling network protocols including wired, wireless and satellite networks with transport protocols such as TCP, and UDP with both unicasting and multicasting capabilities. It models web, Telnet, and FTP applications and also includes the implementation of ad-hoc routing and sensor networks. It provides provision for gathering statistics, tracing, and error modeling for the simulations carried out. Apart from the core code of the NS, there have been a lot of contributions from other researchers, but the majority of these are for terrestrial networks. NS-2 is the standard defacto for terrestrial and ad hoc networks. However, it lacks standard extensions to the satellite environment and it is not widely used in this setting. Only TCP has been implemented in the satellite environment by NS-2 [VKM02, Ana04]. And it is mainly for GEO satellites.

OPNET

OPNET Modeler [Doc00, Doc05] is the industry's leading simulator specialized for network research and development. It allows people to design and study communication networks, devices, protocols, and applications with great flexibility. It provides a graphical editor interface to build models for various network entities from physical layer modulator to application processes. All the components are modeled in an object-oriented approach which gives intuitive easy mapping to real systems. It gives a flexible platform to test new ideas and solutions with a low cost. OPNET is a simulator built on top of a discrete event system. It simulates the system behavior by modeling each event happening in the system and processes it by user-defined processes. Similar to NS-2, it uses a hierarchical strategy to organize all the models to build a whole network. The hierarchy models entities from physical link transceivers, antennas, to CPU running processes to manage queues or running protocols, to devices modeled by nodes with process modules and transceivers, to network models that connect all different kinds of nodes together. TCP over wireless networks has also been simulated over OPNET [ABAR06]. However, OPNET suffers from the same problem of NS-2 regarding satellites and is not widely used in this area.

GeNeSi

The Generic Network Simulator (GeNeSi) is a simulator specially designed for wireless and wired networks developed at the Department of Computer Science, University of Pisa [Nid04]. GeNeSi works not in an event driven model but a hybrid simulation model called *time stepped with refractioning*. In this model the simulation time is split into steps of length d , as in the time stepped model. However, actors of the simulation can ask for a finer grain where needed splitting the actual step. The n -th step is identified by its bounds nd and $(n+1)d$. During an update at step n an actor can request a refraction at some time, let's say $(n+0.3)d$, because this is an important time for the evolution of the actor in the simulation and thus more precise results can be got from the simulation. GeNeSi, like NS-2 and OPNET adopts an object oriented philosophy, however the hierarchy of classes is particularly simple as shown in Figure 5.5.

5.4.2 Simulation setting in GeNeSi

We have chosen to simulate CRT on top of GeNeSi, because GeNeSi already includes many modules for satellite simulation and it is very easy to extend. Our simulation is organized as follows, we create satellite nodes based on the device class module of GeNeSi. Each node contains two identifiers: *SatX* and *SatY*. These two identifiers are used as satellite addresses in simulation. *SatX* is the identifier that declares the

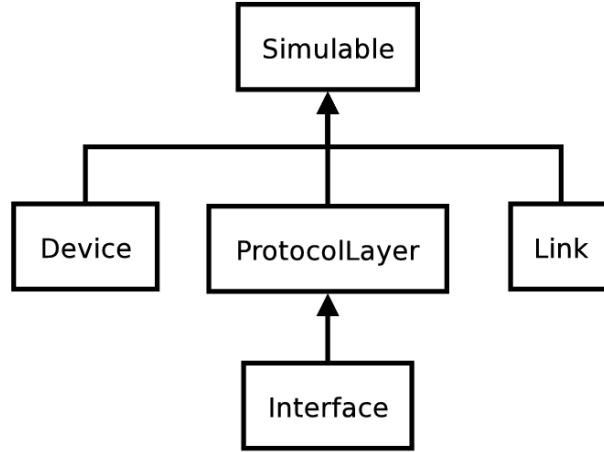


Figure 5.5: The simple class hierarchy of GeNeSi

orbit of the satellite. SatY is the identifier that denotes the position of satellite in the orbit. We simulated four LEO constellation with different sizes:

- 18 satellites; 3 orbits, 6 satellites per orbit
- 36 satellites; 6 orbits, 6 satellites per orbit
- 72 satellites; 6 orbits, 12 satellites per orbit
- 144 satellites; 12 orbits, 12 satellites per orbit

In the simulator, we extended the link module of GeNeSi to emulate ISLs. Links are attached to nodes. There is a duplex link between each pair of neighboring satellites. Each duplex link has $155Mb$ bandwidth in both directions. We simulated the constellation with ATM ISLs which is similar to the structure presented in paper [WDV⁺97, Wer97, WDB97]. Each link has a fixed propagation delay time according to the number of satellites in the constellation. For instance, if the constellation has 18 satellites we set the delay time $0.1s$ for each link. In case of 36 satellites in the constellation, the delay time changes to $0.05s$.

Our CRT algorithm is implemented in the ProtocolLayer module of GeNeSi. We set up the necessary interfaces for ProtocolLayer module to access device and link modules for buffering, transmitting and receiving. We simulate the route of each individual packets. Once a packet is dropped due to congestion, a failure information is written to a log file. Once a packet is transmitted successfully, a success record is written to the log file containing the time stamps of both the beginning and the end of transmission to compute the end-to-end delay time.

Traffic generators We generate two classes of traffic in the simulation. One class contains the traffic in which packets are not strongly related. The other models stream bursts, the main traffic target for our algorithm. We generate traffic using a traffic generator on each satellite. Each generator can produce two kinds of traffic: *stream bursts* and *individual packets*.

- Individual packets: denote those voice phone traffics and Internet services without any special requirements like data for web accessing or file downloading.
- Stream bursts: denote the traffic which may have a strong correlation between arrivals, for instance UDP for video services.

The traffic generator gets seeds from a *traffic matrix*. The traffic matrix is a matrix containing information about global traffic load distribution on the earth. The traffic generator uses the value of vector in the matrix to build traffic. For instance, the generator picks up a value which is 2 from Table 5.5. It uses this value 2 to generate the volume of traffic that equals 20% of the maximum bandwidth which is 155Mb/sec. We first consider a uniform traffic distribution for experiments in this Chapter. A uniform traffic distribution means setting the same value for all seeds in the traffic matrix (Table 5.5 shows the uniform distribution with low traffic value). Three values are chosen for seeds to present different levels of network load: 2 means 20% of maximum bandwidth for light network load, 5 means 50% of bandwidth for medium load and 8 means 80% of bandwidth for high load; once the generator gets a seed s corresponding to the position (SatX, SatY) of the satellite. It generates $s\%$ of bandwidth both for individual packets and bursts. The individual packet generator directly generates numbers of packets with random destinations based on the seed. The stream burst generator generates streams with variable lengths from 250 to 50,000 packets from each satellite to a destination. Each stream consists in a **START** packet at the beginning and a **STOP** packet at the end. In the second series of experiments we change the traffic matrix to a non-uniform distribution (shown in Table 5.6) according to the global traffic demands. This traffic demand model was proposed in [FGM⁺00, FGP02], and represents the actual phone traffic on current constellations.

Comparable algorithms We chose several algorithms to be compared with CRT through simulation. In order to cover different strategies of congestion avoidance. First, we choose Dogleg to represent algorithms that do not take into consideration at all congestion. Then we selected Parallel Highway and Polar Hop because of their non adaptive congestion avoidance rules. Finally, we chose Greedy algorithm as a comparable candidate of adaptive congestion avoidance. We want to compare the behavior of Greedy limited exchanging network information with our global control message exchanging.

[illegible][illegible]

5.4.3 Simulation Results

In this Section, we will discuss the results from several experiments. We select the results of general packets dropping rates and average end-to-end delivery time to evaluate our protocol. We test our protocol in different network scales and also with different time epoch length.

Effect of Network Scales and Network Loads

Our algorithm requires a global exchanging of control messages. The cost for this exchanging is variable depending on the size of the constellation. When the network becomes larger more time is spent to exchange information. Once the new traffic loads are broadcast, each satellite needs to update its global load matrix onboard. In this period, the satellite still uses the old paths to forward the packets. The larger the networks we have, the longer will be the delay generated for matrix updating.

In Figure 5.6, we present the average packets dropping difference between the routed paths by our algorithm and the paths used by Dogleg algorithm, Parallel Highways algorithm and Polar Hop algorithm. The horizontal coordinates denote the number of satellites in the constellation. The vertical coordinates represent the average packets dropping percent for satellites in the constellation. We test these four algorithms under different values of uniform traffic distribution. We chose 20%, 50% and 80% as the the different values for the distribution matrix, which generates low, medium and high network traffic of the bandwidth of the single link.

The experiment shows that, the packets dropping of CRT almost linearly reduced when the network scale becomes large. The reason is that CRT uses as many satellites as possible to balance the traffic load in the constellation. The more satellites in the constellation, the more adaptive routes CRT computes. The results also show that Dogleg, Parallel Highways and Polar Hop behave better than CRT when the constellation is small (18 satellites in the constellation) and the traffic load is low. However, Polar Hop drops lots of packets when there are 36 satellites in the constellation. The reason for this phenomenon is that a constellation with 36 satellites has high congestions in the polar regions. If the number of satellites in the constellation is large enough to allow for alternative routes (about 36 satellites in our simulations) CRT drops less messages than these three algorithms in all the traffic conditions. The results also show that CRT behaves similarly to Greedy in this experiment and we have more results that show the difference between CRT and Greedy later through other experiments.

Another important factor for the performance of the routing algorithm is the effect of the average network traffic load. Figure 5.6 also shows that the network load has a linear influence on CRT, and that in all traffic distributions CRT behaves better than the others.

The experiments in Figure 5.6 is done using the same priority for the **START**

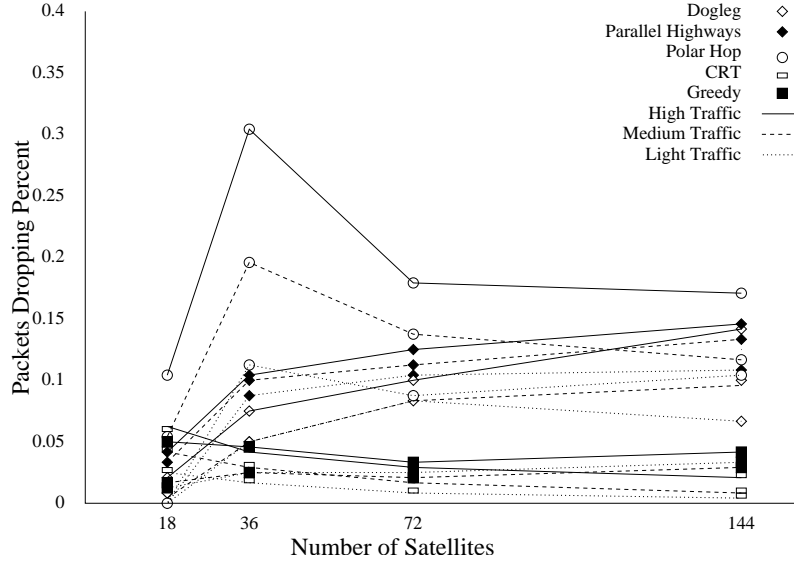


Figure 5.6: Packets Dropping Rates in Different Network Scale and Loads

segments and for normal data. This is not particularly wise as CRT establishes routes using the **START** segments of bursts. Once the **START** of a burst is dropped because of congestion then the remaining packets of the same burst are also dropped. In order to guarantee that all **START** arrive at the destination satellites, we can set a higher priority of transmission for **START** packets than normal packets. We purely test our CRT protocol based on two different assumptions: first without any qualifications, then set higher transmission priority for **START** segments than normal data. The experiment, results in Figure 5.7, shows that if we set the priority, then dropping rates will greatly decrease. This is because in this case we first guarantee the routes establishment, and also the forwarding of the remaining data of the stream. Otherwise once the **START** is lost then the whole transmission will be discarded. The results of simulation show that the dropping rate decreases more than 60%.

Average End to End Delivery Time

The second experiment computes the average time taken by each packet to be delivered from its source to its destination. Our algorithm may choose a non-minimum propagation delay path in case of long queues onboard of a certain satellite. In this experiment, we evaluate the phenomenon measuring the average end-to-end packet delivery time.

In Figure 5.8, we show the average end to end packets delivery time of four algorithms for uniform distribution with different network loads. We still set the

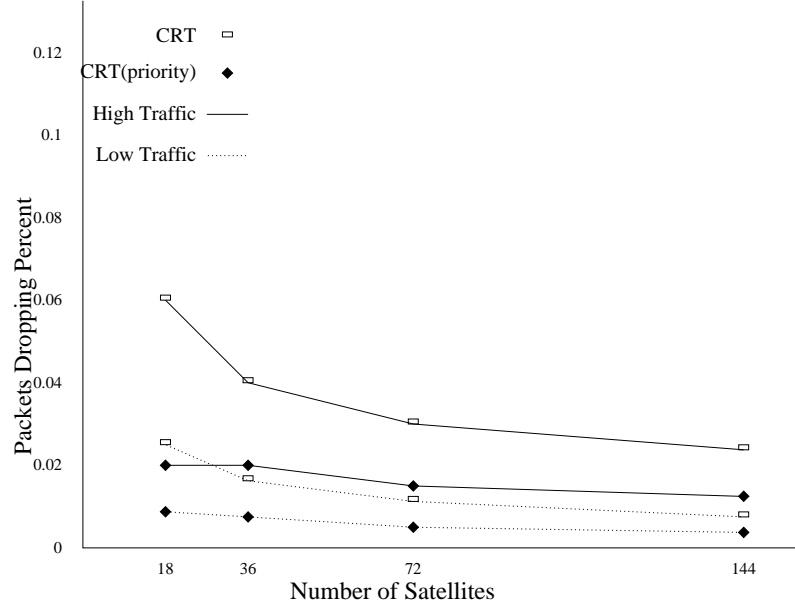


Figure 5.7: Packets Dropping Rates With Priority

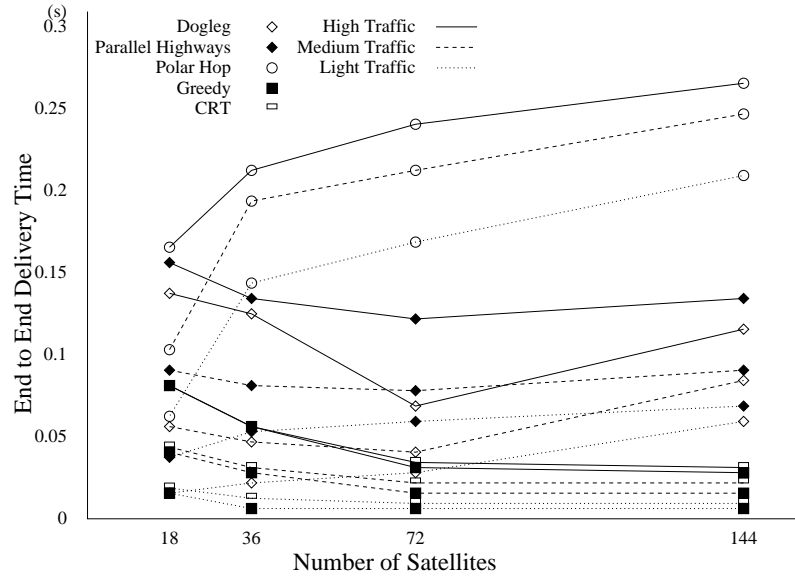


Figure 5.8: End-to-End Delivery Time (Uniform distribution)

horizontal coordinates as the number of satellites in the constellation. The vertical coordinates are the average end-to-end delay time for each packet from the source satellites to the destination satellites. The delay time is computed based on the initial time step insert at the source satellite for each packet and the arrival time of a packet at the destination satellite.

The results show that the Polar Hop algorithm has worst behavior because all the packets are routed to the polar region. In this way, packets always have long propagation delay and it is also easy to have congestion in the polar region. The Dogleg algorithm has good behavior when we have a small number of satellites in case of light network loads. But when the network load increases, the end-to-end delivery time also of Dogleg increases greatly due to congestion. Dogleg also shows quite good behavior corresponding to the number of satellites (72 satellites in our simulation). The Parallel Highway algorithm has a corresponding behavior to the network loads. When the network load increases, the average delay time of Parallel Highways also increases. Our CRT algorithm also has an increasing end-to-end delay time related to the network loads, however among four algorithms, CRT behaves better than the above three because its end-to-end increases less than the others. The reason for the increasing delays is clearly the congestion and route changing. In these cases more time will be spent forwarding in higher network load areas. But there are also some relations between number of satellites and the delivery time. For CRT algorithm, the more satellites that join in the network, the less the time packets will spend from the sources to the destinations. This may be due to the presence of many alternative routes that CRT is able to exploit in the constellation. The results also show that CRT and Greedy have slightly different behaviors in general. Especially when the constellation is small (18 satellites) and the network load is high, the delay of CRT and Greedy is very close. However, when the satellite number increases, Greedy has less delay than CRT.

Different Update Time Epochs

Since the updating of load value plays a very important role in CRT, we should also study the impact of different update time intervals on the overall performance. If the updating happens too often, the bandwidth will be wasted by control messages. On the other hand, if the updating is rare, the routes computed may not be adaptive enough for the traffic conditions.

In this experiment, we compared the performance of CRT with the update interval varying from 0.01s to 0.10s. In Figure 5.9, the horizontal coordinates report different control message update intervals and the vertical coordinates are average packets dropping percents on each satellites. Both *synchronous* updating and *asynchronous* updating are simulated. In synchronous updating mode all satellites broadcast their control messages at the same system time. In asynchronous mode, each satellite starts to compute its own time interval after it receives the latest

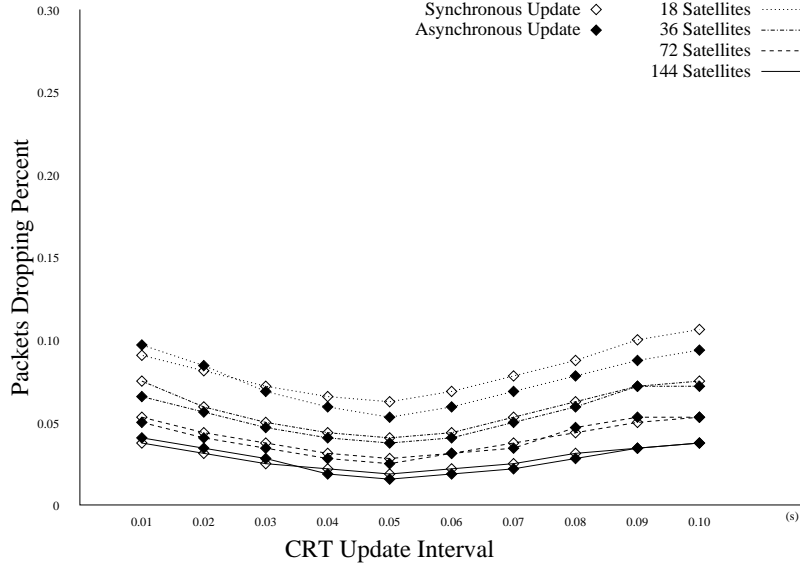


Figure 5.9: Packets Dropping Rates with Different CRT Update Time

control message from all of other satellites for the last time epoch.

From Figure 5.9, we can see that different update intervals have different packet dropping rates. If the updating time is set at 0.01s, the dropping rates are greater than the updating time of 0.05s. The reason is that when we have frequent updates, too many overhead control packets, which have higher priority, will occupy the bandwidth, causing the normal packets to be discarded. When we increase the updating time to 0.10s, we have a high dropped rate of 0.10. This is because the packets dropping is due to congestion as routes are computed by the outdated information. Therefore an appropriate load update interval can greatly reduce the packets dropping rate. On the other hand, the asynchronous updating mode exhibits better behavior especially when there are more satellite nodes in the constellation. In real cases, synchronous updating is simpler than asynchronous updating. The time cost of asynchronous updating is dependent on the delay time of receiving all the updates. Once the updates are dropped because of link errors, fault tolerance is needed. For instance, setting an updating time threshold can be helpful to overcome link error problems. In synchronous updating we can simply setup a local timeout to solve faults. In our experiments we used a fixed synchronous updating time of 0.05s as the default which appears to have the best tradeoff.

Buffering

In this experiment, we mainly focus on the influence of buffer size onboard satellites on the performance of CRT. We compare the performance of CRT with the one of Greedy algorithm in different scenarios of buffer size.

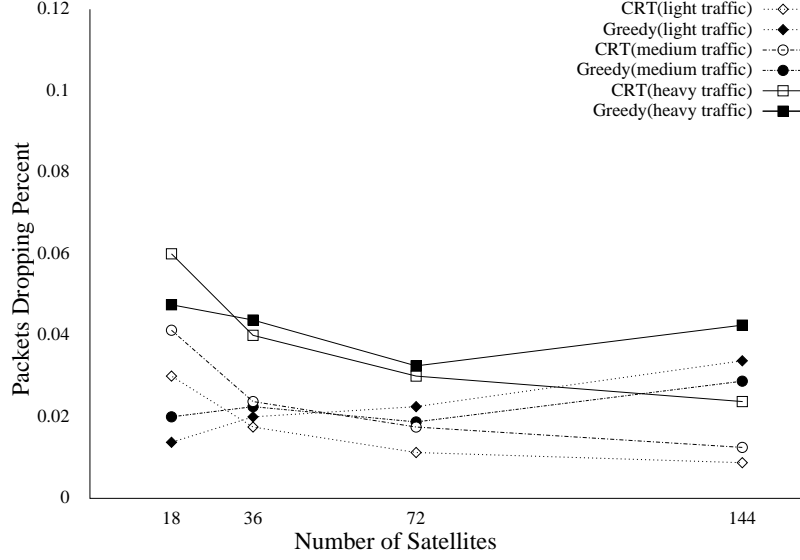


Figure 5.10: Packets Dropping Percents CRT and Greedy without buffering

We evaluate the average packets dropping percents in this test. As in the first experiment, we set the horizontal coordinates as the satellite numbers in the constellation and the vertical coordinates represent the average packets dropping percents. Both CRT and Greedy are tested in different uniform traffic distributions. The buffer spaces of satellites are created in the device class of GeNeSi. We set four different buffer sizes onboard each satellite for the experiment: $0kB$, $128kB$, $256kB$ and $512kB$.

Figure 5.10 shows the results of implementing CRT and Greedy with no buffer space. The same as the results in the first experiment, CRT greatly reduces the packets dropping when the number of satellites in the constellation increases. Greedy only has a better behavior than CRT when the number of satellites is very few (18 satellites in our experiment). This is because CRT do not have many alternative routes since the number satellites is limited. If the constellation has more than 36 satellites then CRT works better than Greedy.

Figure 5.11, 5.12 and 5.13 show the results of the test with buffer size of $128kB$, $256kB$ and $512kB$. Comparing the results in Figure 5.11 and Figure 5.10, we can see the buffer space is helpful both for CRT and Greedy to reduce the packet dropping phenomenon. Moreover, CRT improves more than Greedy with this $128kB$ buffer size. This is because more satellites participate in forwarding packets in CRT. More buffers are used for packet queueing. Thus packets dropping because of congestion is avoided. In Figure 5.12 we can see CRT improves corresponding to the buffer space onboard satellite. Especially in Figure 5.13, we zoom in the vertical coordinates to make the results clear. When the buffer size is large enough ($512kB$ in our test),

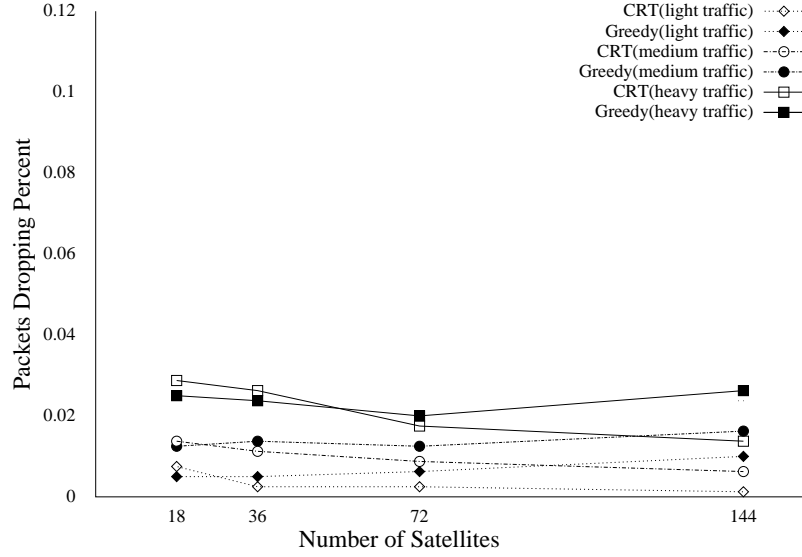


Figure 5.11: Packets Dropping Percents CRT and Greedy with 128kB buffer

CRT can achieve 0 packet dropping.

Non-uniform traffic distribution

Now we change the simulation environment to a non-uniform traffic distribution using the traffic matrix in Table 5.6. We still compare CRT with the other four algorithms in the packet dropping rate and end-to-end delay time.

Figure 5.14 shows the results of packet dropping rates. The results report that CRT behaves better than the others algorithms in most cases except when the constellation is small with only 18 satellites. This phenomenon proves that CRT provides a better balancing mechanism than other algorithms. Moreover, CRT works better if more satellites participate in routing in the constellation .

The results in Figure 5.15 report the end-to-end delay time under non-uniform distribution. It is clear that Greedy has a better performance than CRT under non-uniform traffic condition. The last two experiments show that CRT and Greedy have a different emphasis. CRT provides services with a higher quality and Greedy makes a service faster. However, it must be underlined that the end-to-end time is measured only on the packets that successfully arrive at their destinations. All the packets that are dropped are not included. However, packets dropped lower the QoS and must be rerouted. Thus, we can expect that in many applications CRT exhibits a better tradeoff between end-to-end delivery and the number of packets dropped.

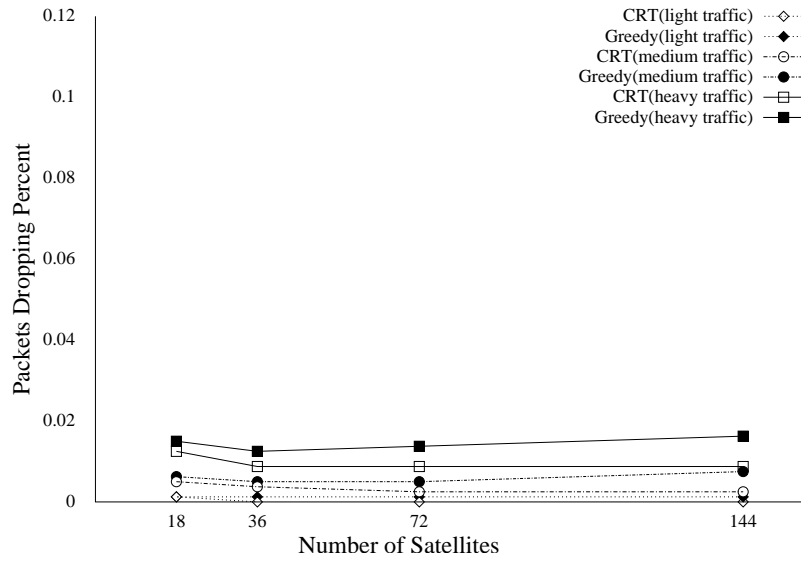


Figure 5.12: Packets Dropping Percents CRT and Greedy with 256kB buffer

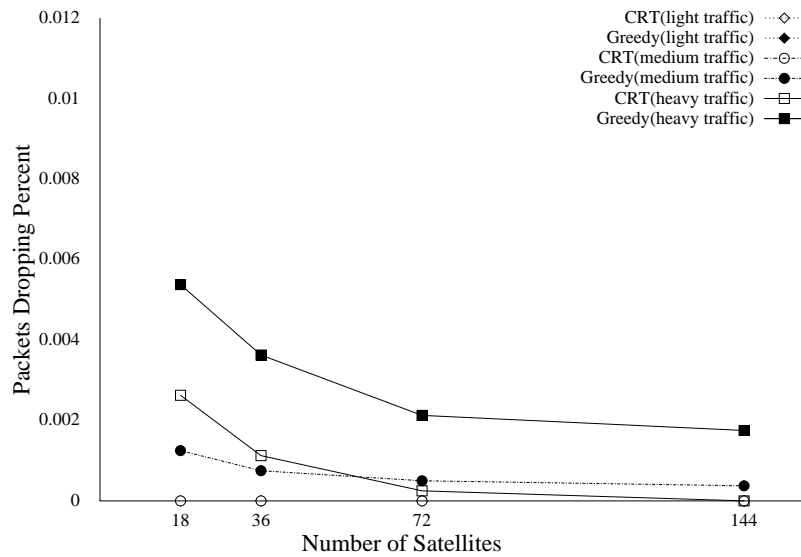


Figure 5.13: Packets Dropping Percents CRT and Greedy with 512kB buffer

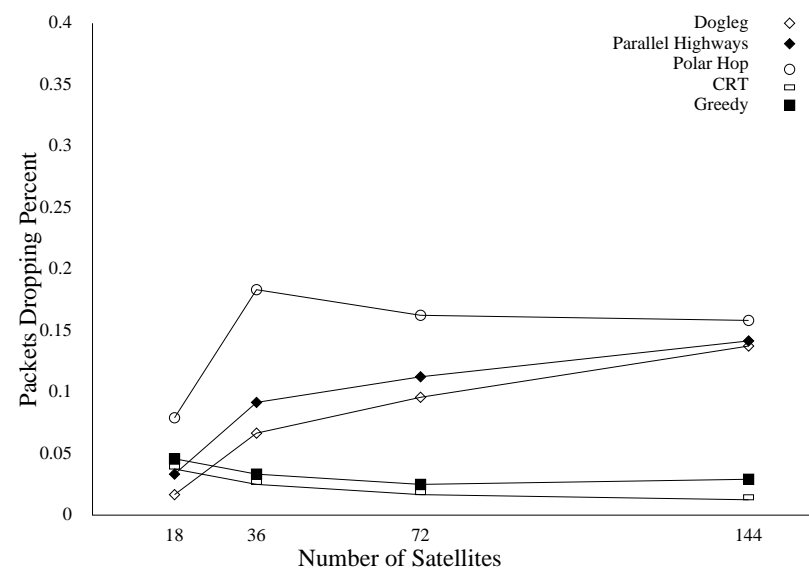


Figure 5.14: Packets Dropping Rates (Non-uniform distribution)

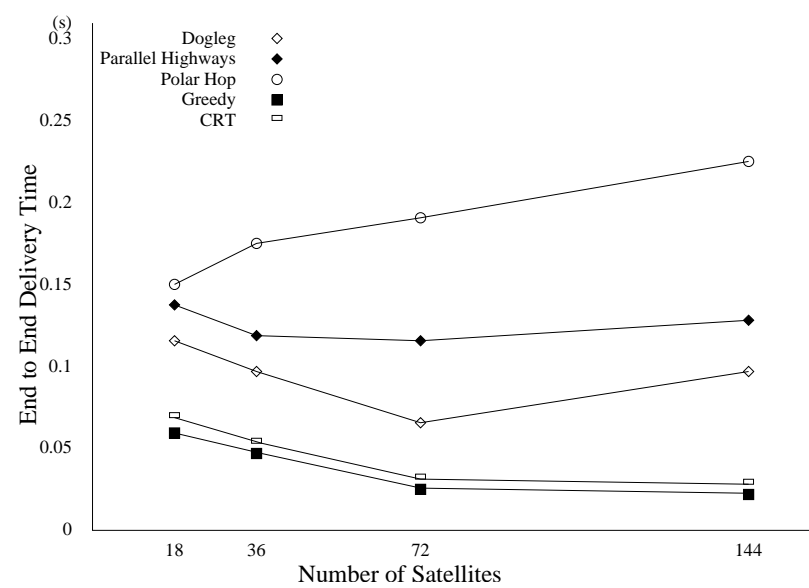


Figure 5.15: End-to-End Delivery Time (Non-uniform distribution)

5.5 Summary

In this chapter, We have described our CRT protocol. CRT protocol has been simulated extensively and its behavior has been compared with previously proposed routing algorithms under different traffic conditions. The results presented show that CRT behaves better than the other algorithms in all traffic conditions. In particular, CRT works much better when the traffic load is heavy. Our results also illustrate that CRT behaves much better on average end to end delivery time. CRT also improves its behavior when there is enough buffer space onboard satellite. But all the results above are based on both uniform traffic distribution and a world wide traffic demand distribution with pure stream message traffic. More experiments should be done with different kinds of traffic together. Also onboard rerouting should be taken into account in the next step of simulation. In the next chapter, we will examine a new version of CRT protocol with bandwidth allocation capability (BCRT). Simulation of BCRT will include onboard buffer size and other properties like priority transmission.

Chapter 6

CRT with priority and bandwidth request (BCRT)

6.1 Motivation

CRT is essentially a the best effort routing algorithm, which accommodates both long bursts of related packets and sporadic short messages traveling on the constellation. As discussed in Section 5.3, the simulations show that the algorithm manages to exploit alternative routes in presence of congestion, achieving a small drop rate and a competitive end-to-end delivery time in all traffic conditions. However, there are the categories of Internet traffic for which it is important to know (and be able to trust on) the quality of the service provided (QoS). We focus particularly on CBR and VBR internet services. CBR requires a constant amount of bandwidth to be devoted and guaranteed for the burst during the delivery. VBR requires the guarantees for a variable amount of bandwidth during its delivery. In this case, once one of these services has been admitted in, the QoS can be measured as the ability of the system to actually deliver the promised bandwidth.

In Section 4.4, the TCD algorithm was discussed [SMK⁺04], which addresses the problem of bandwidth guarantees on the satellites. In TCD the different classes of services are considered, and the different objective functions are used to compute routes for each class of service. For instance, in TCD the residual bandwidth on ISLs is utilized as the routing function for service with high QoS like videos. The minimum propagation delay is used as the key function to compute routes for services without strong QoS requirements, such as web services. The main disadvantages of TCD is that it is very complex because it should distinguish services and lead the contention between the routes computed by different functions, which should be taken into account.

Our main idea is to investigate whether CRT can be simply extended to take

into account of QoS for CBR services.

It is clear that plain CRT is not able to fulfill the requirements of CBR traffic. As soon as the packets arrive, they are delivered, and if the congestion arises they are simply dropped. In order to deal with the bandwidth guarantees, we must be able to *reserve* some bandwidth for a given message for a certain period of time. In this way, when a message request arrives, the system can check whether it can be *admitted* in the constellation with a certain bandwidth guarantee k by actually trying to reserve it. If the reservation has been successful, then the promise can be fulfilled, otherwise the request can be denied or delayed, depending on the policy agreed with the user.

In the rest of the chapter, we extend CRT to a new version of the protocol, *Bandwidth CRT* or BCRT, which supports bandwidth allocation and re-routing. We start providing the constant bandwidth reservation for the whole burst, which can be exploited by CBR services. Then we briefly discuss how the solution can be extended to VBR. A VBR with variable requirements can be regarded as a sequence of CBRs. Moreover, the transcoding between VBR and CBR can be done at the base stations.

Bandwidth (and route) reservation in BCRT is basically done during the initial opening, when the **START** segment traverses the route to the destination satellite. In this phase, each satellite makes a temporary bandwidth reservation for this service request. Then, if the reservation succeeds on a whole path from source to destination, the request is admitted and the data are transmitted.

BCRT still works in time epoches and keeps track of all the active bandwidth reservation made on all links. In the rest of the chapter, BCRT is discussed in detail, the its performance is presented through simulations.

6.2 CRT with bandwidth allocation (BCRT)

In this section, the BCRT algorithm is described particularly. As happens in plain CRT, BCRT assumes the same constellation architecture and models the constellation using a weighted directed graph in which nodes represents the satellites ,and each directed edge denotes a (directional) ISL (Section 5.2). The constellation is a two-dimensional regular mesh. An example is shown in Figure 6.1, BCRT is different from CRT which weights the edges with all the bandwidth reservation currently holding for ISL (that is used by a connection passing through the ISL). For instance, ISL (4, 3) is labeled with 3 active reservations. These three reservations require 7, 9, and 9 units of bandwidth. The total required bandwidth of 25 units. Each unit of bandwidth can correspond to different MBps, depending on the specific used ISL. It is assumed that all the ISL are uniform in our constellation and that the units of bandwidth available on each ISL must be less than a fixed value B or equal to a fixed value B .

A request for CBR burst transmission from the Earth, is performed by sending a request header to the source satellite. The header includes the destination satellite (as in CRT) plus an extra parameter that we will denote with λ , which specifies the units of bandwidth needed by the transmission. λ is computed by the base stations, depending on the characteristics of the service. It is related to the number of bits that should be forwarded in the unit of time to satisfy with the QoS for the service. To simplify the computation, here we only regard λ as abstract unit of bandwidth.

Upon the receipt of a BCRT request, the source satellite starts a preliminary phases to decide whether the request can be fulfilled or not. It actually executes a distributed protocol to decide if there is a path from the source to the destination, on which all ISLs have at least a λ spare bandwidth to reserve for the new message. This is done using **START** packets as in CRT with a different protocol that will be explained in detail in Section 6.2.1. At the end of this reservation phase, the source satellite receives an answer from the queried satellites which either agrees upon the reservation or denies it.

BCRT works in time epochs and uses a CM matrix to hold the information about the state of the constellation. Both the usage and the update of CM are the same as CRT. The main difference is the fact that the value of CM related to a given ISL denotes the units of bandwidth already reserved on that ISL. It is at the beginning of each time epoch and CM is updated with

$$CM_{(i,j)} = \sum_{k=1}^n \lambda_k \quad (6.1)$$

where (i, j) denote the sender and receiver satellite IDs of the ISL, n is the number of active transmissions and λ_k is the number of units of bandwidth required by transmission k .

Once control messages have been exchanged at the beginning of the epoch, CM is updated and each satellite computes the shortest path from itself to all the others using an adjacency matrix as in CRT. In this case, the shortest path is the path where the sum of all the reserved bandwidth is minimal. The results are recorded in SPT again.

6.2.1 Reservation protocol

Let's consider a burst request p with a bandwidth λ_p from source s to destination d . Before agreeing upon transmission, the source satellite s generates a **START** segment that not only contains unique identifier $uid(p)$ and the best route currently

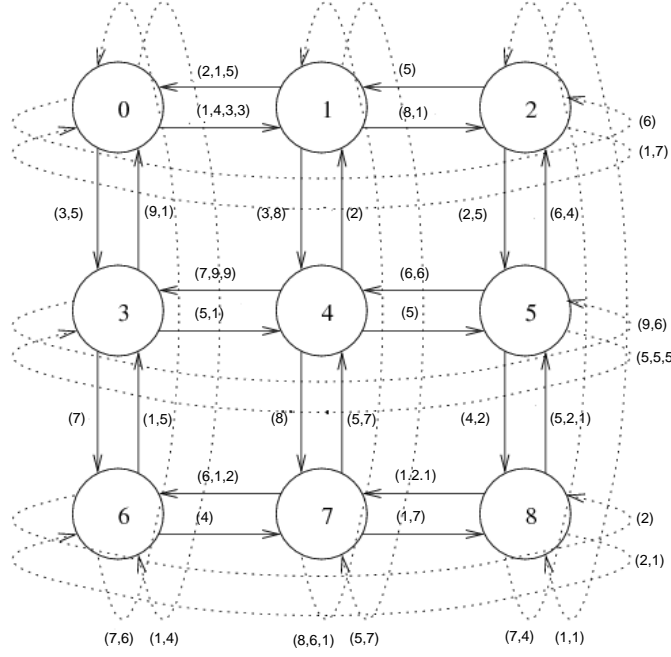


Figure 6.1: 3 × 3 Mesh with bandwidth reservation information

recorded in SPT ($route_p = r_1 = s, r_2, \dots, r_k = d$) but also indicate the bandwidth required (λ_p).

The idea is that we try first to reserve the bandwidth along the current shortest path that is along the less used route. If it does not work, the source tries the alternative routes starting with the remaining three neighbors.

Let's consider now the first attempt, that is the one along the shortest path. Before sending the **START** packet out, s checks the fact that the first outgoing link in the path (s, r_1) has got enough bandwidth available for $uid(p)$, that is

$$CM_{s,r_1} + \lambda_p < B$$

if it is not the case, the attempt is aborted and the source tries a different neighbor satellite x . In particular, it computes the shortest path from s to d fixing x as the first node in the path, and makes a new reservation attempt along the new route.

Otherwise, CM_{s,r_1} is updated adding λ_p reserved units and the **START** packet is forwarded to the next satellite r_1 .

When a generic satellite r_i in $route_p$ receives a **START** reservation packet, it in turn checks if the current real traffic conditions allow the reservation of λ_p units for $uid(p)$.

$$CM_{r_i,r_{i+1}} + \lambda_p < B$$

if it is the case, the node r_i records the message in the routing (inserting $uid(p)$, r_{i+1} , r_{i-1} , λ_p to deal with congested routes that will be explained in a moment) and

marks the entry as ‘to-be-confirmed’. Then, the **START** packet is delivered to the next node in the route r_{i+1} . Satellite r_{i+1} will answer to r_i after having checked the rest of the route. If the answer is positive (bandwidth available on all the routes) the routing entry corresponding to $uid(p)$ will be confirmed and a positive answer will be recorded in the **START** packet for r_{i+1} as part of the confirmed route $route'_p$. Otherwise, r_i will consider the ISL to r_{i+1} as ‘congested’ and attempt rerouting on at most two neighbor satellites (r_{i-1} from which the request has been received is not considered). If no alternative route can be found r_i sends back a negative answer to all previous nodes before r_i , and the $uid(p)$ entry is deleted from the routing table. If a route is found starting from satellite x , x is recorded in table and $route'_p$ in the **START** packet. When the destination d receives the **START** packet, it will send a positive answer back the source s along $route'_p$. When r_i in $route'_p$ receives the positive answer, it will confirm the ‘to-be-confirmed’ entry of $uid(p)$, and $CM_{i,i+1}$ will be updated with the value of λ_p . The source s will start to transmit the normal data of burst p after it receives the positive answer from d . All the packets in the burst will be delivered along the same route $route'_p$. At the end of the burst p , s generates a **STOP** packet like in CRT to signal the end of the burst for all nodes in the route. When r_i in $route'_p$ receives **STOP** with $uid(p)$, it will first release λ_p bandwidth units corresponding to the entry in the routing table and update $CM_{i,i+1}$, then delete the entry.

In order to be able to manage the positive and negative answers and the occupied bandwidth, more information is needed in the routing table entry with respect to plain CRT. Four new entries are introduced to the algorithm, $uid(p)$, r_{i+1} , r_{i-1} and λ_p . r_{i+1} is used to forward the packets and r_{i-1} is used to forward the positive and negative answers reverse back to s . λ_p is used to release the bandwidth units allocated for burst p .

Figure 6.2 shows the flow chart of how BCRT protocol works.

6.2.2 An example with BCRT

It is assumed once more that there is a new burst from satellite 0 to satellite 5 in Figure 6.1. The new burst has a bandwidth requirement $\lambda = 5$ for the service. From the final all-pairs shortest paths table (Table. 6.1, we optimize the table that -1 denotes the direct link otherwise numbers denotes the intermediate node ID), the best path from 0 to 5 is (0,2,5). Thus the burst is processed as follows. Node 0 generates a unique identifier $uid = 93684900$ for the burst, creates a **START** packet containing (93684900, (2,5), 5) and propagates it to node 2, where (2, 5) is the route and 5 is the value of λ . Then, CM is updated locally increasing $CM[0][west]$ by 5 and the pair (93684900, 2, 5) is inserted in the routing table. When the **START** packet is received, the satellites 2 inserts the pair (93684900, 0, 5, 5) in its respective routing tables as a ‘to-be-confirmed’ route. The satellite 2 also updates the congestion matrix increasing $CM[2][south]$ by 5. When the satellite 5 receives the **START** packet, it sends a positive answer back the satellite 0 via the satellite 2. Those ‘to-

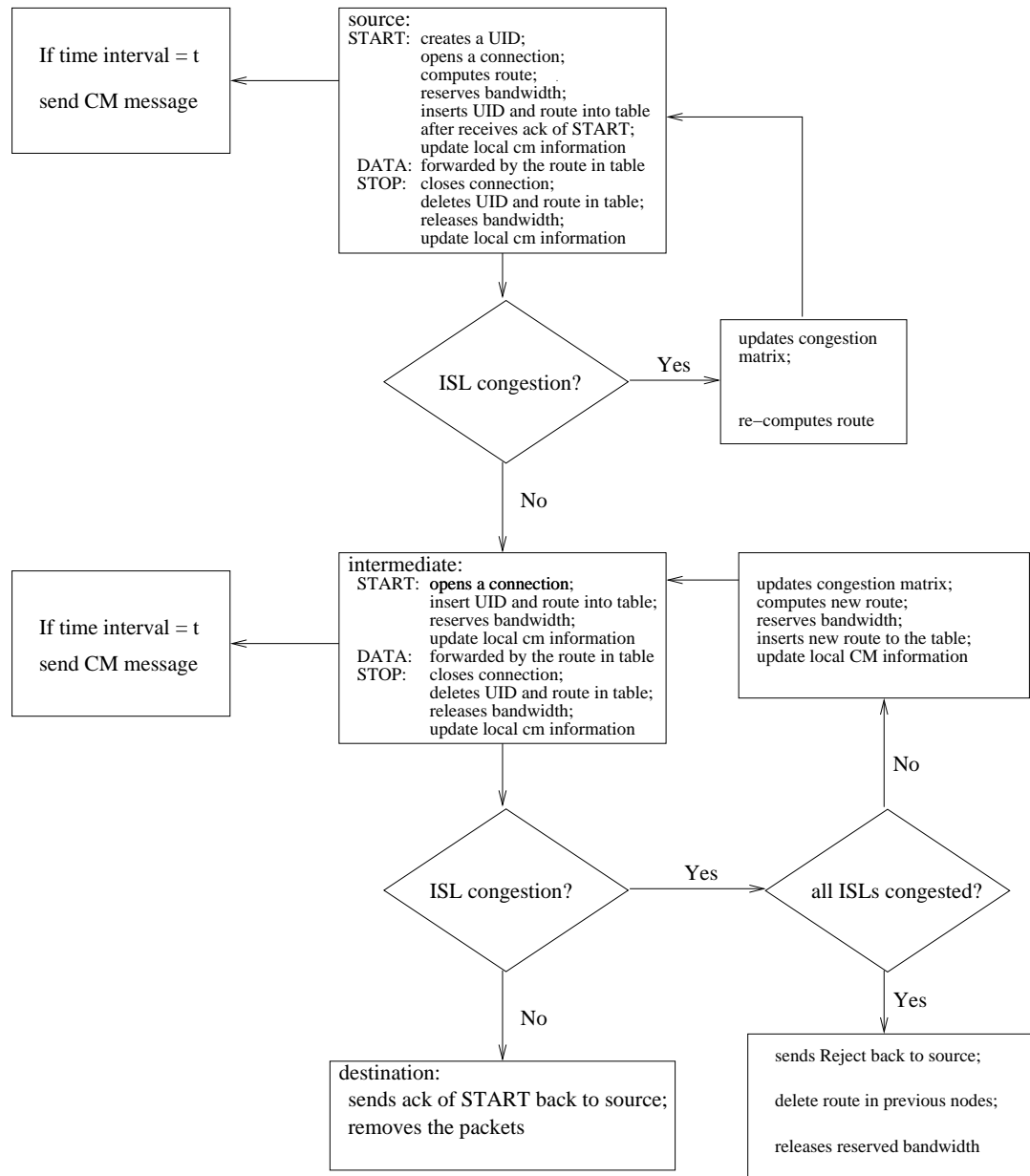


Figure 6.2: Flow chart of CRT protocol

Table 6.1: Shortest paths

-1	-1	-1	-1	3	2	-1	6	6
-1	-1	-1	0	-1	2	0	-1	2
-1	-1	-1	8	1	-1	8	8	-1
-1	4	0	-1	-1	4	-1	6	6
1	-1	1	1	-1	-1	8	-1	5
2	4	-1	-1	-1	-1	8	8	-1
-1	4	8	-1	3	8	-1	-1	-1
4	4	8	6	-1	8	-1	-1	-1
6	2	-1	6	6	-1	-1	-1	-1

be-confirmed' routes in routing tables are confirmed by this positive answer. Thus the transmission of the burst is guaranteed by this bandwidth reservation.

When a data packet is received with $uid = 936849900$, all nodes use the routing table to process it to its destination. At the end of burst, the source satellite creates a **STOP** packet with $uid = 936849900$ and deletes the corresponding entry from the routing table. Also $CM[0][west]$ is decremented by 5. When a **stop** packet is received, the satellite 2 updates its routing tables and reduces $CM[2][south]$ by 5.

6.3 Simulation Setting

The simulation setting for BCRT is not much different from the one presented in Section 5.4.2 for CRT. Both uniform and non-uniform traffic distribution matrixes are used (Table 5.5 and 5.6). The traffic generator used for BCRT has a new parameter λ that denotes the bandwidth requirement. The function of generating a burst is similar to the one introduced in section 5.4.2 except that the **START** which generated at the beginning of a burst which has a private variable λ . λ is randomly generated and it is less than a system threshold which we set to 10.

The BCRT is compared with plain CRT, Greedy and Greedy with re-routing function. The purpose of choosing CRT and Greedy is to observe if BCRT can reduce packets dropping caused by congestion, as compared with these two algorithms. Moreover, Greedy is expanded Greedy with a rerouting after congestion which is called Greedy(R). The rerouting function of Greedy(R) is very similar to that of BCRT. When an ISL is congested, the node will compute a new route base on the other two ISLs except the one which receives the request. And the behavior of packets dropping rate and end-to-end delivery time of entire bursts with rerouting is compared by comparing BCRT with Greedy(R).

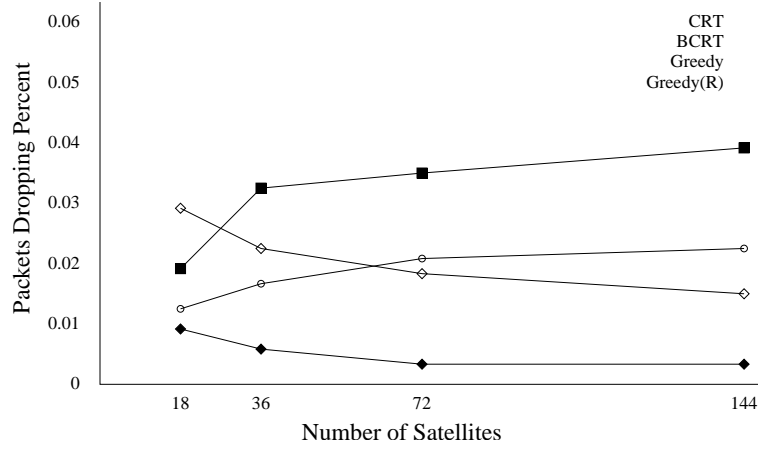


Figure 6.3: Packets Dropping Rates with uniform light Traffic

6.4 Simulation Results

Two series of experiments of BCRT are discussed in this section. The uniform traffic distribution is used in the first series of tests. BCRT is compared with plain CRT, plain Greedy and Greedy(R) in both phase of packets dropping rates and end-to-end delivery time. In the second series of tests the non-uniform traffic distribution matrix is used (Table 5.6).

6.4.1 Experiments with uniform traffic distribution

BCRT reroutes the **START** packets to find an alternative route after congestion. This mechanism guarantees that more bursts will be forwarded to the destination without loss, as compared with CRT.

Figure 6.3, 6.4 and 6.5 show the results of packets dropping percent in the uniform traffic distribution. The average packets dropping difference among BCRT, CRT, Greedy and Greedy(R) is presented. As usual, we set the horizontal coordinates as the number of satellites in the constellation and the vertical coordinates representing the packets dropping percent. We still choose 20%, 50% and 80% as the values for the uniform traffic distribution matrixes to generate low, medium and high network traffic of the bandwidth of a single link.

The results in Figure 6.3, 6.4 and 6.5 show that the drops packet number of BCRT is less than that of all the other algorithms because of congestion. It is proved that the rerouting function of BCRT during the **START** phase can support more bursts to be admitted by the constellation and improve performance. Moreover, if BCRT from CRT is compared with Greedy(R) from Greedy for the improvements, it is found that when the traffic increases BCRT still works better than CRT, while Greedy(R) behaves very close to Greedy. The reason is that BCRT

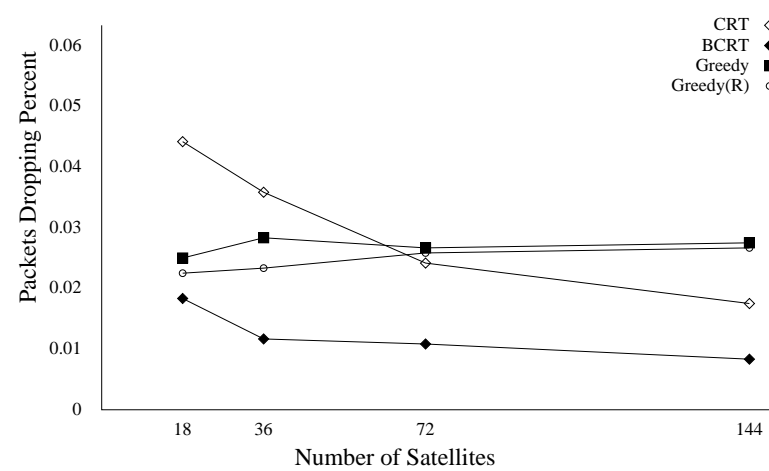


Figure 6.4: Packets Dropping Rates with uniform medium Traffic

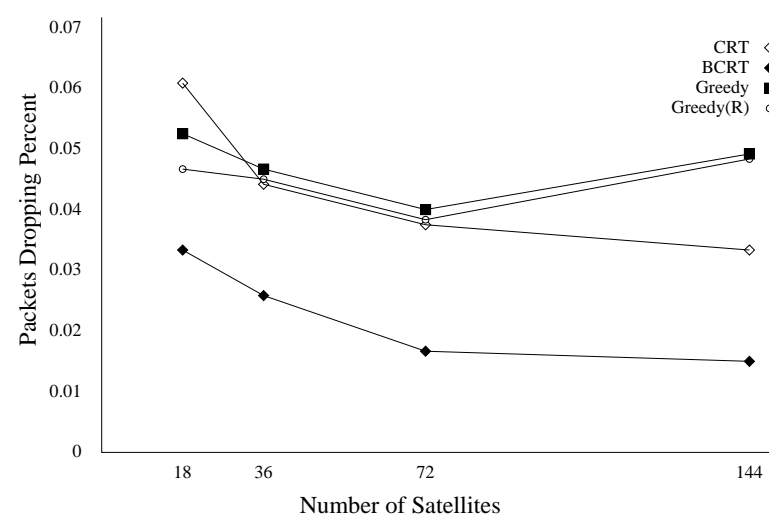
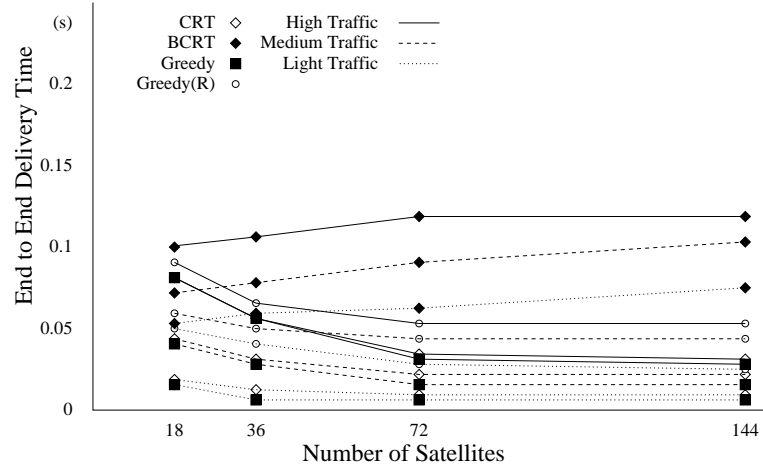
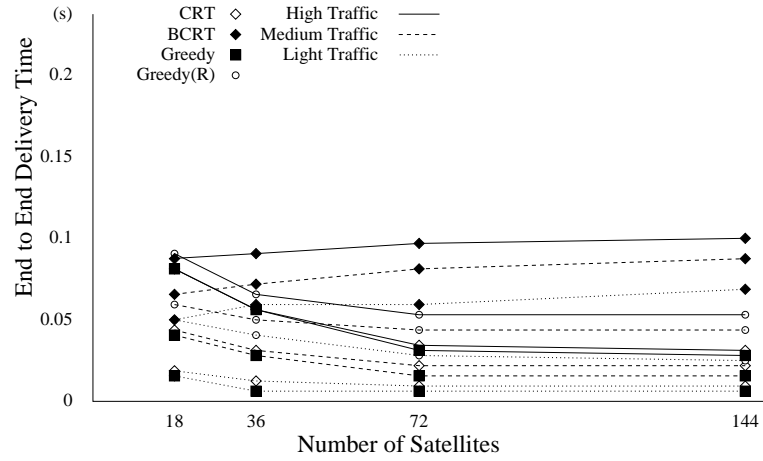


Figure 6.5: Packets Dropping Rates with uniform high Traffic

Figure 6.6: End-to-end delivery time with uniform Traffic(with **START** phase)Figure 6.7: End-to-end delivery time with uniform Traffic (without **START** phase)

reroutes bursts recomputing updated CM matrix while Greedy(R) just simply finds other non-congested ISL to the neighbors.

On the other hand, BCRT re-routes packets after congestions and spends more time to transmit the whole bursts due to the longer ‘**START**’ phase. In the second test of this series, we took into account the behavior of BCRT on end-to-end delivery time. Figure 6.6 and 6.7 show the average end-to-end packets delivery time of BCRT, CRT, Greedy and Greedy algorithm with re-routing. The horizontal coordinates in the figure are the numbers of satellites in the constellation. The vertical coordinates are the average end to end delay time for each packet from their source satellites to the destination satellites. We only considered those packets which arrived at the destinations, not including the dropped ones. The results of BCRT in Figure 6.6 include the time cost of the **START** packets for bandwidth reservation. Figure 6.7 only contains the delivery time of normal packets of BCRT.

The results show that the delay time of BCRT is larger than that of CRT and Greedy in general, because the plain CRT and Greedy have no reaction on the packets dropped by congestion. BCRT spends time to recompute routes for the bursts after congestion. It is found that the delivery time increases when the constellation becomes larger. The reason is that more than one congestion may happen in the route from the source to the destination, bursts may need to be rerouted more than one time. The behavior of the delivery time is also relative to the traffic load. When the traffic load increases, the average end-to-end delivery time of BCRT also increases. Other interesting phenomenon is that the behavior of BCRT is worse than that of Greedy(R), because Greedy only takes into account the congestions about neighbor satellites. The reason is that Greedy(R) strategy makes the creation of hot spot areas easier and the brought congestion of Greedy(R) is more than that of BCRT. When the results of BCRT in Figure 6.6 and 6.7 are compared, it is found that the heavier is the traffic the bigger is the behavior difference, because the **START** phase spends longer time to establish the routes and reserve the bandwidth for bursts.

6.4.2 Experiments with non-uniform traffic distribution

In this series of simulation, we implement the same measurements of packets dropping and end-to-end delivery time but under non-uniform traffic distribution.

Figure 6.8 shows the results of packets dropping percents of BCRT, CRT, Greedy and Greedy(R). BCRT still works better than the others as it behaves under uniform traffic distribution. The behavior of the end-to-end delay time of BCRT and other algorithms is shown in Figure 6.9. BCRT has a quite stable behavior around 0.1s epochs in our experiment. The results illustrate that even the distribution is non-uniform, BCRT drops less packets but has longer end-to-end delivery time.

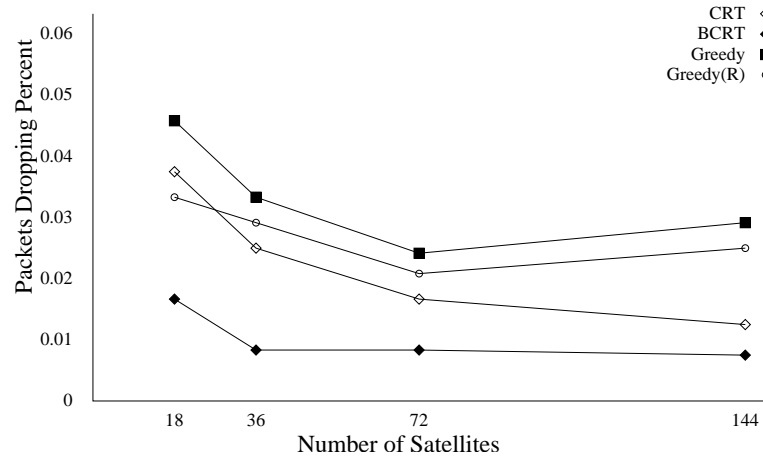


Figure 6.8: Packets Dropping Rates with non-uniform Traffic

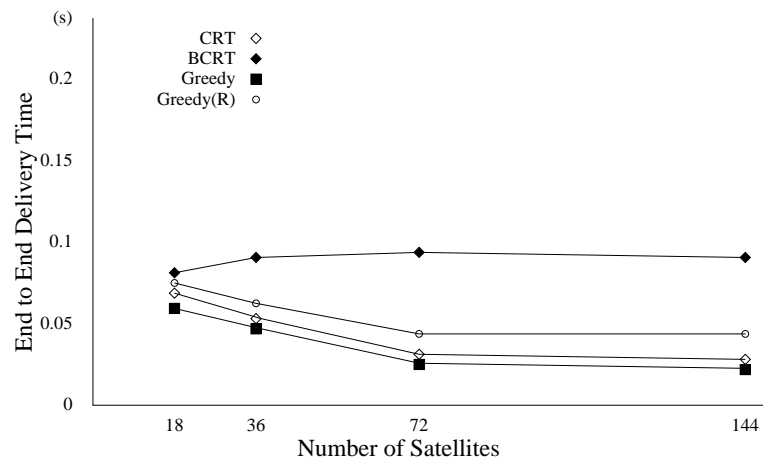


Figure 6.9: End-to-end delivery time with non-uniform Traffic

Buffering

In this series of experiments, we carefully study the relation between routing performance and buffer onboard LEO satellites. We concentrate on three different aspects. The first one is the buffer size on board satellites. The second is the buffering timeout for each burst in the buffer. The third one is using buffering information as a part of parameters to compute routes.

Buffer usage The buffer is mainly used to store the queue of **START** requests of bursts. This is because a transmission tunnel is established between the source/destination pair once the bandwidth is successfully reserved for a burst. Data of the burst will be forwarded without caching except link/hardware errors. Thus, in this series, we only simulate the queue of **START** packets in the buffer.

Buffering timeout BCRT is designed for video bursts. The delay of response time of a service is very important. The **START** packets can not be cached in the buffer for a long period of time. A soft timeout is necessary for each **START** packet in the buffer to control the delay time of the reservation request. We set different number of time epochs as soft timeout thresholds in our simulation.

Routing with buffer information BCRT computes routes using a congestion matrix constructed by the number of active transmissions on ISLs. In this experiment, we explore how information on buffered requests can be useful for routes computation. The request queued in buffer represent possible hot spot areas, thus taking them into account may be more accurate than using only the info in CM about already reserved routes. Fig. 6.10 shows the average end-to-end delay time of each **START** packet under different settings. The vertical coordinates are the average end to end delay time for each **START** packet from their source satellites to the destination satellites. We set the buffer size from $0Kb$ up to $256Kb$ and the buffering timeout from 2 to 3 time epochs(0.15s or 0.2s).

Fig. 6.11 shows the packets dropping rates of all bursts using different buffer size. In this figure, we zoom in the vertical coordinates to make the results clearer. The results of both figures illustrate that buffer space is helpful to reduce the packets dropping because more **START** requests can arrive at the final destination. But the improvement is not linear according to the buffer size. When the buffer size becomes too large the improvement is descendent. We also see that soft timeouts in the queue reduce the end-to-end reservation time at the price of increasing packet dropping rates.

System throughput

System throughput of the whole network is also calculated in this series of experiments. Fig. 6.12 shows the system throughput value of CRT, BCRT, Greedy and

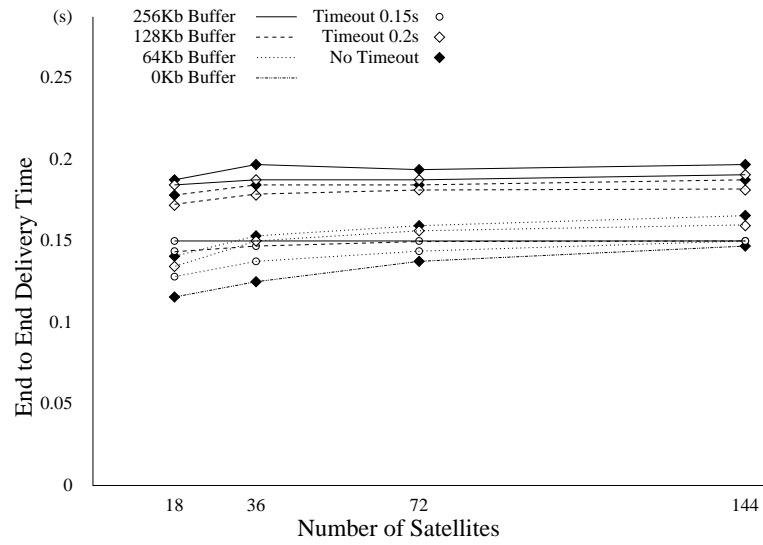
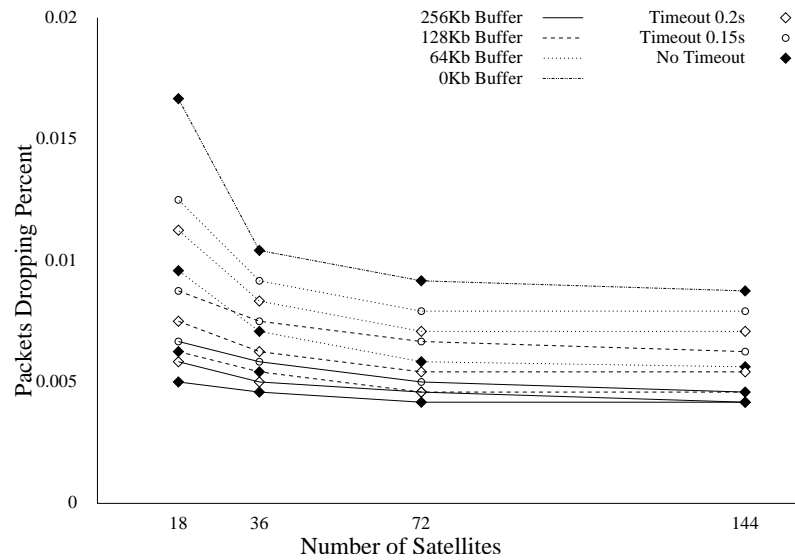
Figure 6.10: End-to-end delivery time of **START** (non-uniform)

Figure 6.11: Packets Dropping Rates with buffering (non-uniform)

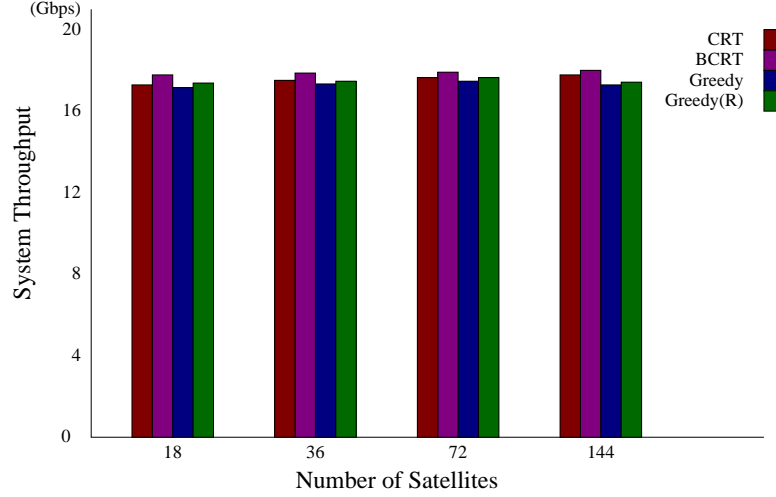


Figure 6.12: System throughput with non-uniform Traffic

	18 Satellites	36 Satellites	72 Satellites	144 Satellites
Time Cost	0.001997	0.002003	0.002010	0.002054

Table 6.2: Time Cost For SP Computing

Greedy(R). The vertical coordinates in the figure is the value of the average system throughput. The results show that BCRT has better behavior than other three in all condition.

6.4.3 Time cost for routes computation

Other critical issue to evaluate a routing protocol is the time cost for computing the routes. The core of both CRT and BCRT algorithms is computation of the congestion matrix. The main time cost of computation of our algorithm is also composed of two parts: the computing the adjacency matrix from the congestion matrix, and the applying the all-pairs shortest paths algorithm.

The time cost of BCRT in different sizes of constellations is recorded. The results of experiments are shown in figure 6.13 and table 6.2, which show that to both of those two computing stages spend very small amount of time together, which equals around 2 milliseconds in our simulation. The time cost of CRT remains almost the same no matter how many satellites in the constellation.

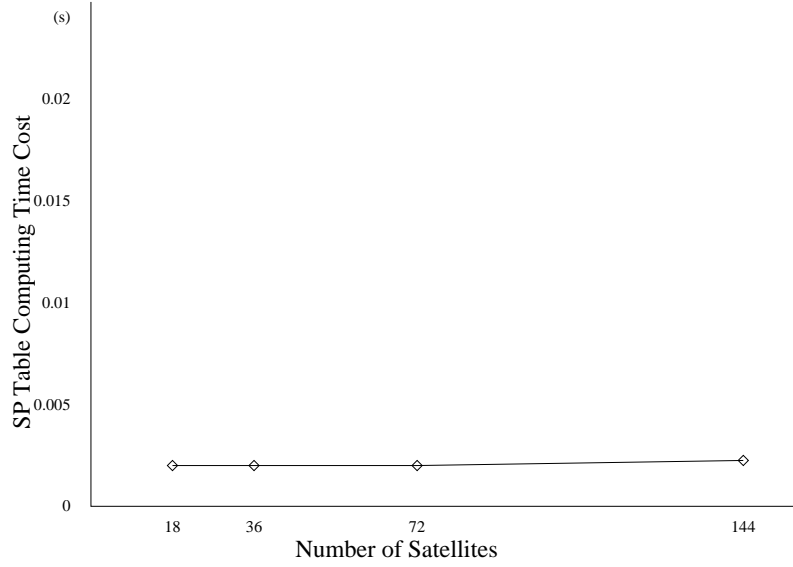


Figure 6.13: Time Cost For Computing SP Table

6.5 Summary

In this chapter, we have introduced the new version of CRT protocol with the bandwidth allocation and rerouting after congestion. BCRT uses an explicit request λ contained in the **START** packet for allocation. Positive answers are sent back the sources after successful bandwidth reservation. More routing information is stored in routing table onboard to forward the answer and allocate the bandwidth. The results of simulations show that BCRT drops few packets. The payment of entire transmission is that BCRT has a longer delivery time than CRT because of rerouting. The simulation shows that using the buffer space can reduce the packet loss of BCRT. The experiment also shows that the computation of BCRT is very cheap. BCRT has very good behavior for stream traffic. In the future, it could be interesting to investigate other aspects, e.g. the reaction to the link failures.

Chapter 7

Conclusion

In this thesis, first of all, the motivation of using LEO satellite constellations for Internet was outlined. Secondly, the technical challenges solved in different protocol layers, from lower physical layer up to transport layer, for integrating LEO constellation with Internet was examined. In particular, we focused on the problem of supporting bursts traffic in LEO constellations. Thirdly, the issues related to the network layer and to the routing problem were discussed. The routing problem is the main topic of this thesis. The traditional Internet routing can not be directly used in LEO constellations. The mobility and long delay time of satellites make Internet routing more severe. The main solution is to isolate LEO constellation as an extra AS and to implement private simple routing protocol inside the constellations. In the literature, different objective functions have been used to solve the routing problem. However, so far no definitive solution has been proposed as many algorithms are not adaptive, and require complex computation.

After discussing the state of the art, we first proposed CRT, a routing algorithm for LEO satellite networks which aims at burst traffic. In CRT, control messages are used to compute routes taking into account network load conditions. CRT works in time epochs and chooses routes by applying a shortest path algorithm on top of a congestion matrix. The algorithm provides congestion avoidance by periodically updating control messages. Then, CRT was evaluated with the simulation experiment, and its results were discussed. CRT was compared with previously proposed routing algorithms under different traffic conditions. The results showed that the behavior of CRT is better than that of the other algorithms in all traffic conditions. In particular, CRT works much better than the others when traffic load is heavy. The results also illustrated that CRT has a impressively better behavior on the average of end-to-end delivery time. Moreover, CRT was evaluated in consideration of buffering. The results indicated that CRT worked better with larger buffer space.

In this thesis we also proposed a second protocol: BCRT, which is a extension of CRT. BCRT takes into account rerouting and bandwidth allocation to provide QoS to the final user. BCRT allows bandwidth reservation along a route. BCRT also works in time epochs. It constructs the control message and congestion matrix

using the sum of used bandwidth on a single ISL. BCRT reroutes bursts in case of congestion. BCRT was simulated under the same conditions of CRT. The results showed that BCRT drops less packets than CRT. On the other hand, the end-to-end delivery time of BCRT is longer than CRT. But the behavior of delivery time of BCRT is still comparable and it is also closely related with network load. BCRT has also been simulated with buffering. The packet dropping rates are greatly reduced by using large buffer space.

An important issue for future work is the fault tolerance. In this thesis, we have only taken into account rerouting for congestions. ISL errors and hardware failures are not considered. How to distinguish about congestion and hardware failure? Which kind of information are needed to recompute new routes after failure? These problems are all in the next step of our research. Moreover, currently BCRT only supports fixed bandwidth reservation. How to support variable bandwidth for VBR is another possible direction for future work. Finally we plan to investigate BCRT for multicast as a promising direction for our research.

Bibliography

- [AAC⁺03] I. F. Akyildiz, O. B. Akan, C. Chen, J. Fang, and W. Su. InterPlaNetary Internet: state-of-the-art and research challenges. *Computer Networks*, 43:75–112, 2003.
- [ABAR06] M. N. Akhtar, M. A. O. Barry, and H. S. Al-Raweshidy. Modified Tahoe TCP for Wireless Networks Using OPNET Simulator. In *Proc. London Communications Symposium 2006*, September 2006.
- [AFP02] M. Allman, S. Floyd, and C. Partridge. Increasing TCP’s Initial Window. Technical report, RFC 3390, October 2002.
- [AGS99] M. Allman, D. Glover, and L. Sanchez. Enhancing TCP Over Satellite Channels using Standard Mechanisms. Technical report, IETF RFC 2488, January 1999.
- [Ahm06] S. Ahmadi. Real-Time Transport Protocol (RTP) Payload Format for the Variable-Rate Multimode Wideband (VMR-WB) Extension Audio Codec. Technical report, RFC 4424, February 2006.
- [AJ97] I. F. Akyildiz and S. Jeong. Satellite ATM Networks: A Survey. *IEEE Communications Magazine*, 35(7):30–44, July 1997.
- [Alo06] Aloha simulation validation. Website: <http://www.laynetworks.com/Aloha%Simulation%Validation.htm>, 2006.
- [AMP01] I. F. Akyildiz, G. Morabito, and S. Palazzo. TCP-Peach: A New Congestion Control Scheme for Satellite IP Networks. *IEEE/ACM Transactions on Networking*, 9(3):307–321, June 2001.
- [Ana04] D. Anantharaman. Performance Analysis of Snack in Satellite Networks Through Simulation. Master’s thesis, Lamar University, 2004.
- [Bal97] A. Ballardie. Core Based Trees (CBT) Multicast Routing Architecture. Technical report, RFC 2201, September 1997.

- [BBC⁺98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. Technical report, RFC 2475, December 1998.
- [BCS94] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. Technical report, RFC 1633, July 1994.
- [BKG⁺01] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby. Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations. Technical report, RFC 3135, June 2001.
- [BZB⁺97] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. Technical report, RFC 2205, September 1997.
- [CE05] C. Chen and E. Ekici. A Routing Protocol for Hierarchical LEO/MEO Satellite IP Networks. *ACM/Kluwer Wireless Networks Journal*, 11(4):507–521, 2005.
- [CEA02] C. Chen, E. Ekici, and I. F. Akyildiz. Satellite Grouping and Routing Protocol for LEO/MEO Satellite IP Networks. In *Proc. of the 5th ACM international workshop on Wireless mobile multimedia*, pages 109–116, 2002.
- [CGM⁺01] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang. TCP westwood: Bandwidth estimation for enhanced transport to over wireless links. In *Proc. MOBICOM 2001, Rome, Italy*, July 2001.
- [CM03] L. Cheng and I. Marsic. Modeling and Prediction of Session Throughput of Constant Bit Rate Streams in Wireless Data Networks. In *Proc. IEEE Wireless Communications and Networking*, volume 3, pages 1733–1741, 2003.
- [Doc00] OPNET Modeler On Line Document. Product Documentation[EB/OL]. Technical report, OPNET Technologies,Inc, 2000.
- [Doc05] OPNET Modeler On Line Document. Product Documentation,Wireless Module User Guider[EB/OL](2005-03-05). Technical report, OPNET Technologies,Inc, 2005.
- [EAB01] E. Ekici, I. F. Akyildiz, and M. D. Bender. Network Layer Integration of Terrestrial and Satellite IP Networks over BGP-S. In *Proc. of GLOBECOM 2001*, volume 4, pages 2698–2702, November 2001.
- [EBA01] E. Ekici, M. D. Bender, and I. F. Akyildiz. A Distributed Routing Algorithm for Datagram Traffic in LEO Satellite Networks. *IEEE/ACM Transactions on Networking*, 9(2):137–147, April 2001.

- [EHT⁺98] D. Estrin, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification. Technical report, RFC 2362, June 1998.
- [EKDT00] Ö. Ercetin, S. Krishnamurthy, S. Dao, and L. Tassiulas. A Predictive OoS Routing Scheme for Broadband Low Earth Orbit Satellite Networks. In *Proc. The 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2000, (PIMRC 2000)*, volume 2, pages 1064–1074, September 2000.
- [FCM00] P. Fraise, B. Coulomb, and B. Monteuis. SkyBridge LEO Satellite: Optimized for Broadband Communications in the 21st Century. In *Proc. 2000 IEEE Aerospace Conference*, volume 1, pages 241–251, March 2000.
- [Fen97] W. Fenner. Internet Group Management Protocol, Version 2. Technical report, RFC 2236, November 1997.
- [FGM⁺00] A. Ferreira, J. Galtier, P. Mahey, G. Mateus, A. Oliveira, and M. Pichereau. An Optimization Model for Routing in Low Earth Orbit Satellite Constellations. In *Proc. of ISPAN*, pages 161–166, December 2000.
- [FGP02] A. Ferreira, J. Galtier, and P. Penna. Topological design, routing and hand-over in satellite networks. In I. Stojmenovic, editor, *Handbook of Wireless Networks and Mobile Computing*, pages 473–507. John Wiley and Sons, 2002.
- [FLC01] J. F. Frigon, V. C. M. Leung, and H. Chan Bun Chan. Dynamic Reservation TDMA Protocol for Wireless ATM Networks. *IEEE Journal on Selected Areas in Communications*, 19(2):370–383, Feb 2001.
- [FLYV93] V. Fuller, T. Li, J. Yu, and K. Varadhan. Classless Inter-Domain Routing (CIDR):an Address Assignment and Aggregation Strategy. Technical report, RFC 1519, 1993.
- [Gav97] B. Gavish. LEO/MEO systems - Global mobile communication systems. *Telecommunication Systems*, 8(2-4):99–141, December 1997.
- [Hal97] S. Halabi. *Internet Routing Architectures*. Cisco Press, second edition, 1997.
- [HK00] T. R. Henderson and R. H. Katz. On Distributed, Geographic-Based Packet Routing for LEO Satellite Networks. In *Proc. IEEE Global Telecommunications Conference 2000*, volume 2, pages 1119–1123, December 2000.

- [HL01] Y. Hu and V. O. K. Li. Satellite-Based Internet: A Tutorial. *IEEE Communications Magazine*, 39(3):154–162, 2001.
- [HMK98] A. Hung, M. Montpetit, and G. Kesidis. ATM via satellite: A framework and implementation. *Wireless Networks*, 4:141–153, 1998.
- [HP06] Y. He and S. Pelagatti. CRT: an Adaptive Routing Protocol for LEO Satellite Networks. In *Proc. 2nd IEEE International Conference On Information and Communication Technologies: from Theory to Applications, Damascus, Syria, 24-28 April 2006*.
- [HS98] Y. Hashimoto and B. Sarikaya. Design of IP-based Routing in a LEO Satellite Network. In *Proc. of Third International Workshop on Satellite-Based Information Services (WOSBIS '98)*, pages 81–88, October 1998.
- [HYK04] Q. Huang, B. S. Yeo, and P. Y. Kong. A Routing Algorithm to Provide End-to-End Delay Guarantee in Low Earth Orbit Satellite Networks. In *Proc. IEEE 59th Vehicular Technology Conference*, volume 5, pages 2911–2915, May 2004.
- [HYK05] Q. Huang, B. S. Yeo, and P. Y. Kong. An Enhanced QoS Routing Algorithm for Provision of End-to-End Delay Guarantee in Low Earth Orbit Satellite Networks. In *Proc. 2005 IEEE Wireless Communications and Networking Conference*, volume 3, pages 1485–1490, March 2005.
- [JP93] Y. G. Jan and K. M. Peterson. Us5268694: Communication system employing spectrum reuse on a spherical surface. Technical report, Motorola, Granted Motorola patents related to Iridium, 1993.
- [KE03] R. Kucukates and C. Ersoy. High Performance Routing in a LEO Satellite Network. In *Proc. The Eighth IEEE International Symposium on Computers and Communications (ISCC'03), Kemer-Antalya, Turkey*, pages 1403–1408, 2003.
- [KM99] O. Kubbar and H. T. Mouftah. An Aloha-Based Channel Access Scheme Investigation for Broadband Wireless Networks. In *Proc. The Fourth IEEE Symposium on Computers and Communications*, page 203, 1999.
- [Kri90] K. R. Krishnan. Markov Decision Algorithms for Dynamic Routing. *IEEE Communications Magazine*, pages 66–69, Oct 1990.
- [KT99] H. Koraitim and S. Tohmé. Resource Allocation and Connection Admission Control in Satellite Networks. *IEEE Journal on Selected Areas in Communications*, 17(2):360–372, Feb 1999.

- [LABJ00] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet Routing Convergence. In *Proc. ACM Special Interest Group on Data Communications Conference SIGCOMM'00, Stockholm, Sweden*, pages 175–187, September 2000.
- [LC00] Y. J. Lee and T. Christian. Laser satellite communication: precise estimation of the satellite position and synchronization and the stabilization of the beam on the receiving satellite. In *Proc. of the IEEE 2000 National Aerospace and Electronics Conference, 2000. NAECON 2000*, pages 696–703, October 2000.
- [Li03] W. Li. Inter-Domain Routing: Problems and Solutions. Technical report, TR-128, Technical Reports, Experimental Computer Systems Lab, Stony Brook University, 2003.
- [LLKK00] J. W. Lee, J. W. Lee, T. W. Kim, and D. U. Kim. Satellite over Satellite (SOS) Network: A Novel Concept of Hierarchical Architecture and Routing in Satellite Network. In *Proc. of 25th Annual IEEE Conference on Local Computer Networks (LCN 2000.)*, pages 392–399, November 2000.
- [LMG93] R. J. Leopold, A. Miller, and J. L. Grubb. The iridium system: a new paradigm in personal communications. *Applied Microwave and Wireless*, 5(4):68–78, 1993.
- [LMJ98] C. Labovitz, G. R. Malan, and F. Jahanian. Internet routing instability. *IEEE/ACM Transactions on Networking*, 6(5):515–528, 1998.
- [LMYW04] D. Li, X. Mao, J. Yu, and G. Wang. A Destruction-Resistant Dynamic Routing Algorithm for LEO/MEO Satellite Networks. In *Proc. The Fourth International Conference on Computer and Information Technology (CIT'04), Wuhan, China*, pages 522–527, Sept 2004.
- [LSGS04] M. Luglio, M. Y. Sanadidi, M. Gerla, and J. Stepanek. On-Board Satellite "Split TCP" Proxy. *IEEE Journal of Selected Areas in Communications*, 22(2):362–370, February 2004.
- [Mar05] F. Martelli. *Packet Scheduling Algorithms for Interconnected Single-hop Multichannel Systems*. PhD thesis, University of Pisa, 2005.
- [Moy94] J. Moy. Multicast Extensions to OSPF. Technical report, RFC 1584, March 1994.
- [MR97] R. Mauger and C. Rosenberg. QoS Guarantee for Multimedia Services on a TDMA-Based Satellite Network. *IEEE Communications Magazine*, 35(7):56–65, July 1997.

- [MWSK02] M. Mohorcic, M. Werner, A. Svigelj, and G. Kandus. Adaptive Routing for Packet-Oriented Intersatellite Link Networks: Performance in Various Traffic Scenarios. *IEEE Transactions on Wireless Communications*, 1(4):808–818, October 2002.
- [NBBB98] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. Technical report, RFC 2474, December 1998.
- [NCD98] P. Narvaez, A. Clerget, and W. Dabbous. Internet Routing over LEO Satellite Constellations. In *Proc. of the Third ACM/IEEE International Workshop on Satellite-Based Information Services(WOSBIS'98)*, pages 89–95, October 1998.
- [Ngu03] H. N. Nguyen. *Routing and Quality-of-Service in Broadband LEO Satellite Network*. Kluwer Academic Publishers, MA, USA, 2003.
- [Nid04] F. Nidito. Hitchhikers' Guide To The GeNeSi Framework Programming. Department of Computer Science, University of Pisa, 2004.
- [Örs98] T. Örs. *Traffic and Congestion Control for ATM over Satellite to provide QoS*. Phd thesis, Centre for Communication Systems Research, School of Electronic Engineering, IT and Mathematics, University of Surrey, United Kingdom, December 1998.
- [Pos80] J. Postel. User Datagram Protocol. Technical report, RFC 768, August 1980.
- [Qui01] B. Quinn. IP Multicast Applications: Challenges and Solutions. Technical report, RFC 3170, September 2001.
- [Rag02] S. Raghavan. An In-Depth Design Guide to ATM over Satellite Communication Networks. In *Proc. ECPE 6504: Wireless Networks and Mobile Computing Individual Project Report*, 2002.
- [RE00] M. M. Riad and M. M. Elsokkary. Optimal Algorithms for Routing in LEO Satellite Networks with ISL. *HiPC 2000, Springer*, 1970/2000:489–500, 2000.
- [RF99] K. Ramakrishnan and S. Floyd. A Proposal to Add Explicit Congestion Notification (ECN) to IP. Technical report, RFC 2481, January 1999.
- [RNJ03] K. Ramakrishnan, K. R. Namuduri, and S. K. Jayaweera. Multi-priority video partitioning for CDMA based communication systems. In *Proc. IEEE 58th Vehicular Technology Conference*, volume 5, pages 3405–3409, Oct 2003.

- [SC02] J. W. So and D. H. Cho. Access Control of Data in Integrated Voice/Data/Video CDMA Systems. In *Proc. IEEE 55th Vehicular Technology Conference*, volume 3, pages 1512–1516, 2002.
- [SFCJ04] K. Scott, P. Feighery, B. Crow, and M. Jurik. TCP Congestion Control in Shared Satellite Environments. 2004.
- [SG96] M. A. Sturza and F. Ghazvinian. The Teledesic Satellite System. In *Proc. 4th Budapest International Conference on Up-to-Date Satellite Communications*, September 1996.
- [SG99] S. Shimer and R. Gray. High bandwidth optical intersatellite link technologies. In *Proc. of International Topical Meeting on Microwave Photonics, Melbourne, Australia*, volume 1, pages 101–104, 1999.
- [SM04] J. Sun and E. Modiano. Routing Strategies for Maximizing Throughput in LEO Satellite Networks. *IEEE Journal On Selected Areas In Communications*, 22(2):273–286, Feb 2004.
- [SM06a] H. P. E. Stern and S. A. Mahmoud. Theory of Code Division Multiplexing. <http://zone.ni.com/devzone/conceptd.nsf/webmain/58B2834DE12D473C86256E00007AAFB5>, 2006.
- [SM06b] H. P. E. Stern and S. A. Mahmoud. Theory of Frequency Division Multiplexing. <http://zone.ni.com/devzone/conceptd.nsf/webmain/A3AB58D0D470AC2686256E000079BBDC>, 2006.
- [SM06c] H. P. E. Stern and S. A. Mahmoud. Theory of Time Division Multiplexing. <http://zone.ni.com/devzone/conceptd.nsf/webmain/7E02B98203BDD20586256E000077EC24>, 2006.
- [SMK⁺04] A. Svigelj, M. Mohorcic, G. Kandus, A. Kos, M. Pustisek, and J. Bester. Routing in ISL Networks Considering Empirical IP Traffic. *IEEE Journal On Selected Areas In Communications*, 22(2):261–272, Feb 2004.
- [SO97] P. L. Spector and J. H. Olson. Application of SkyBridge LLC for authority to launch and operate the Skybridge system. *filing with the US Federal Communications Commission, SkyBridge LLC*, February 1997.
- [TJKN05] T. Taleb, A. Jamalipour, N. Kato, and Y. Nemoto. IP Traffic Load Distribution in N GEO Broadband Satellite Networks. *Computer and Information Sciences - ISCIS 2005, Springer*, 3733/2005:113–123, 2005.
- [TKW04] C. W. Tzeng, K. W. Ke, and H. T. Wu. Resource Allocation and Adaptive Routing in Multimedia Low Earth Orbit Satellite Mobile Networks.

- In *Proc. 2004 IEEE International Conference on Multimedia and Exop (ICME)*, pages 1795–1798, 2004.
- [UP06] Researchers At UC Berkeley LBL USC/ISI and Xerox PARC. The ns Manual (formerly ns Notes and Documentation). Technical report, The VINT Project, A Collaboration between UC Berkeley, LBL, USC/ISI, and Xerox PARC, November 2006.
- [VCG98] C. Villamizar, R. Chandra, and R. Govindan. BGP Route Flap Dampening. Technical report, RFC 2439, 1998.
- [VKM02] D. Velenis, D. Kalogeras, and M. Maglaris. Sa(tpep): a TCP Performance Enhancing Proxy for Satellite Links. In *Proc. 2nd International IFIP-TC6 Networking Conference, Pisa Italy*, May 2002.
- [WC96] Z. Wang and J. Crowcroft. Quality-of-service routing for supporting multimedia applications. *IEEE Journal of Selected Areas in Communications*, 14(7):1228–1234, September 1996.
- [WD88] D. Waitzman and S. Deering. Distance Vector Multicast Routing Protocol. Technical report, RFC 1075, November 1988.
- [WDB97] M. Werner, C. Delucchi, and K. Burchard. ATM Networking for Future ISL-based LEO satellite constellations. In *Proc. of the International Mobile Satellite Conference 1997 (IMSC '97), Pasadena, California*, pages 295–300, June 1997.
- [WDV⁺97] M. Werner, C. Delucchi, H. Vogel, G. Maral, and J. D. Ridder. ATM-Based Routing in LEO/MEO Satellite Networks with Intersatellite Links. *IEEE Journal on Selected Areas in Communications*, 15(1):69–82, January 1997.
- [Wer97] M. Werner. A Dynamic Routing Concept for ATM-Based Satellite Personal Communication Networks. *IEEE Journal on Selected Areas in Communications*, 15(8):1636–1648, October 1997.
- [Woo01a] L. Wood. *Internetworking with satellite constellations*. PhD thesis, School of Electronics, Computing and Mathematics, University of Surrey, 2001.
- [Woo01b] L. Wood. IP Routing Issues In Satellite Constellation Networks. *International Journal of Satellite Communications*, 19(1):69–92, January/February 2001.
- [WYTW06] K. Wang, K. Yi, B. Tian, and C. Wu. Packet routing algorithm for polar orbit LEO satellite constellation network. *Science in China: Series F Information Sciences*, 49(1):103–127, 2006.

- [YZL05] Z. Yuan, J. Zhang, and Z. Liu. A Simplified Routing and Simulating Scheme for the LEO/MEO Two-Layered Satellite Network. *ICCNMC 2005, Springer*, 3619/2005:742–751, 2005.