Routing for Predictable LEO/MEO Multi-Layered Satellite Networks

Heyu Liu and Fuchun Sun

Abstract LEO/MEO Multi-Layered Satellite Network (MLSN), consisting of low and medium earth orbit satellites, is capable of providing higher coverage and better service than most Single-Layered Satellite Network. Its performance, however, has been longly encumbered by obsolete routing protocols and algorithms. This paper takes the predictability of satellite movements into consideration, based on which a novel routing protocol—Predictable Satellite Network Routing Protocol (PSNRP), is proposed. In this protocol, all topology changes due to satellite movement are classified into predictable and unpredictable changes. This predictability assists to reduce the protocol overhead. The simulations show that except for obtaining better routing performance, PSNRP also successfully allocates calculation resources evenly among all nodes, separates user data from protocol control data, and achieves stronger robustness on undergoing satellite failures and link congestions.

Keywords MLSN · LEO/MEO · Routing · Predictable · Protocol · Robustness

1 Introduction

Satellite networks are indispensable in modern communication network, especially in the districts where wired accesses are not available. Satellite networks consist of satellites running on earth orbits with Inter-Satellite Links (ISLs) connecting them.

H. Liu (✉) · F. Sun
State Key Laboratory of Intelligence Technology and Systems, Department of Computer Science and Technology, Tsinghua University, Beijing, People's Republic of China
e-mail: liuheyukicker@yahoo.com.cn

F. Sun
e-mail: fcsun@mail.tsinghua.edu.cn

F. Sun et al. (eds.), *Foundations and Practical Applications of Cognitive Systems and Information Processing*, Advances in Intelligent Systems and Computing 215,
DOI: 10.1007/978-3-642-37835-5_44,

In accordance with their altitudes, orbits are classified into Low Earth Orbit (LEO), Medium Earth Orbit (MEO), and Geostationary Earth Orbit (GEO). MLSN, especially LEO/MEO MLSN is becoming a critical issue in the research of satellite network for its lower end-to-end delay and stronger robustness over Single-Layered Satellite Networks. For every communication network, routing scheme is decisive. Different from the fixed topology in conventional ground network, the rapid time-variant topological change and complicated structure in LEO/MEO satellite network make it complicated to design a proper routing protocol.

Three categories of strategy are proposed to deal with topological change: Virtual Topology, Virtual Node, and Strategies dependent on topology [1]. In Virtual Topology, the whole cycle of satellite network T is divided into n timeslots $[t_0 = 0, t_1)$, $[t_1, t_2)$, …, $[t_{n-1}, t_n = T]$, and the topology is deemed as fixed in every timeslot. In another word, topology changes occur only in time nodes t_1, t_2, …, t_n. Due to the predictability of satellite movement, all the time nodes and their relevant topologies could be off-line pre-calculated. [2] Virtual Topology strategy was usually adopted in the FSA-based routing [3], the ATM-based routing [4] and snapshot-based routing [5]. The Virtual Node topology attempts to divide the surface of the earth into grids, and every grid is represented by a virtual node whom is described by an invariant logical address. According to the strategy, the satellite covering the grid for the longest time is bounded with this node, and it transfers the relevant information to successive satellite during decoupling. The Virtual Node strategy actually shields satellite relative movement to the ground at the cost of greatly increasing on-board processing data. The IP-based routing [6] and distributed routing [7] belong to this strategy. In recent research, Strategies dependent on topology attract growing attention. Satellites are capable of perceiving the real-time topology and opting different routing strategies. Strategies dependent on topology are, however, designed for some specific topologies [8]. Of these three categories, Virtual Topology is the most widely applied in satellite network routing research for its off-line computation ability and utilization of predictable satellite motion [9].

As for the LEO/MEO MLSN, researchers have designed a number of routing protocols. Predictable Link-State Routing (PLSR) proposed by Fischer successfully takes advantage of satellite movement to simplify protocol. However, it is designed for Single-Layered Satellite Network [10]. Multi-Layer Satellite network Routing algorithm (MLSR) is designed for MLSN [11], but it cannot meet QoS requirements. Chen adopted Virtual Topology Grouping strategy instead of Virtual Topology, and first presented the idea of grouping [12]. But in her SGRP protocol, the whole cycle is divided into too many short time slots that a lot of them are not long enough to deploy subsequent routing protocol and algorithms, resulting in topology jitter. To merge time slots, long proposed NSGRP based on an improved Virtual Topology Grouping strategy [13], but it only applies to GEO/MEO/LEO Triple-Layered Satellite Network.

The above protocols are mainly encumbered by two drawbacks: 1. they do not effectively make use of the predictability of satellite movement. Different from the mobile ad-hoc network, the movements of satellites are predictable and imply great values. But these protocols only use it to divide time slots; 2. The robustness of satellite network is very weak on facing emergencies. For example, MEO satellites in NSGRP take charge of all the controlling and most of the routing tasks, which often make them hot-spots. Once anomalies occur in these bottlenecks, the whole performance of network degrades. Meanwhile, re-routing strategy is not efficient in handling system anomalies.

In this paper, we improve the idea of treating the topological change as an array of topology snapshot handoffs [5]. Each topology snapshot describes the fixed network linking status in a time slot, and the handoff order reflects the predictable topological change of the network. Based on this topology model, we propose a Predictable Satellite Network Routing Protocol (PSNRP). Simulation results indicate that PSNRP shows a high performance on end-to-end delay and robustness to anomalies.

The constellation model is discussed in Sect. 2 and then topology model in Sect. 3. PSNRP is illustrated in details in Sect. 4. Simulation results and relevant analysis are depicted in Sect. 5. Finally, Sect. 6 is a brief conclusion and future works.

2 Constellation Model

The MLSN in this paper consists of LEO layer and MEO layer, Table 1 shows the parameters of the constellation. The running cycles in the table are calculated using Kepler equation given the orbit height [14].

LEO. The sub-constellation in LEO layer adopts the Walker star constellation model. It is able to provide continuous and complete coverage of the earth surface.

MEO. The sub-constellation in MEO layer adopts Walker-δ constellation model. It also continuously and completely covers both the earth surface and LEO layer.

Ground Station. In our model, there are two kinds of ground stations—GGs (Ground Gateways) and CCs (Control Centers). GGs are used to switch user data with LEO satellites while CCs take charge of switching control data of the routing protocol with MEO satellites.

Four kinds of duplex links connect satellites and ground stations. They are Inter-Satellite Links (ISLs), Inter-Orbital Links (IOLs), User Data Links (UDLs), and User Control Links (UCLs).

Table 1 Parameters of constellation

	Orbit height (km)	Running cycle (hour/h)	Satellite number	Orbit obliquity
LEO	895.5	12/7	6 × 8	90°
MEO	10390	6	2 × 5	45°

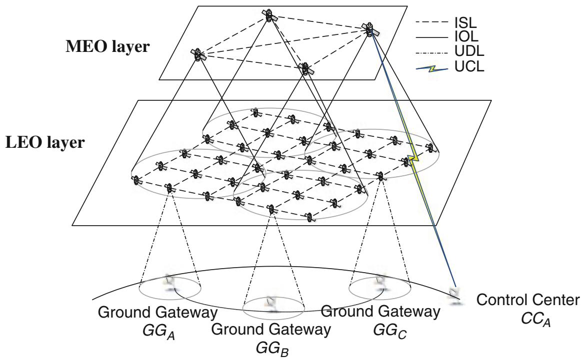


Fig. 1 The structure of a LEO/MEO MLSN

UCLs are first proposed in this paper and they are used to transfer routing protocol control data between CCs and satellites. UCLs are stipulated to exist only between MEO satellites and Control Centers, the reason will be further discussed later.

Figure 1 depicts the LEO/MEO MLSN structure.

3 Topology Model

In this section, a concrete topology model is constructed for LEO/MEO MLSN. The core idea is that the cycle is divided into a number of time slots according to the Virtual Topology Grouping strategy and within a time slot the topology is deemed as fixed. A snapshot is abstracted from a fixed topology to help routing. Given the predictability of network topology, all the snapshots can be calculated previously; therefore, topology change can be deemed as the handoffs of an array of snapshots.

3.1 Snapshots and Division of Time Slot

The definition of topology snapshot was proposed in [5] to describe the mobility for LEO single-layered air-craft network. Now, based on the Virtual Topology Grouping strategy, we formally give the definition of snapshot and division of time slot in LEO/MEO MLSN.

Definition 1 Time slot and time node: a time slot is a short period of time during satellite network running cycle. For $t', t'' \in \mathrm{T}$ with $t' < t''$, $[t', t'')$ is used to represent the time slot $\{t \in T| \; t' \leq t < t''\}$. A time node is a time-point distinguishing two adjacent time slots.

Definition 2 Division of time slot: the moment a LEO satellite leaves a group (see Sect. 4.1 for details of grouping) for another one is defined as a time node, thus a new time slot is generated. In another word, the topology is deemed as fixed if no LEO satellite has changed the group it belongs to. This method for division is proved feasible in [5].

According to this method, the whole running cycle can be divided into n time slots denoted by $[t_0 = 0, t_1)$, $[t_1, t_2)$… $[t_{n-1}, t_n = T)$. The topology in every time slot is deemed as fixed, it changes only in time nodes $t_1, t_2, \ldots, t_n$. This method of dividing time slot lays the fundamental of the Virtual Topology Grouping strategy [15].

Definition 3 Topology snapshot: the fixed topology in any time slot $[t_i, t_{i+1})$ could be described by a weighted digraph $G_i = (V_i, E_i)$, in which V_i and $E_{i,,}$ respectively, represents the nodes and edges in graph G_i. Two-tuple $S_i =< t_i, G_i>$ is defined as the topology snapshot of time slot $[t_i, t_{i+1})$.

Therefore, all the topology changes in a cycle can be replaced by an array of topology snapshots $<S_0, S_1, S_2, \ldots, S_n>$, of which every S_i can be pre-calculated. In practical operation, the protocol usually sets a mapping function Φ. $S_{i+1} = \Phi(S_i)$ describes the relationship between current snapshot and the next one.

3.2 Predictable Change and Unpredictable Change

In this topology model, predictable changes equal to the handoffs of an array of topology snapshots. The predictability of snapshots makes it possible to cut down OBP overheads. Handling the unpredictable changes caused by anomalies in MLSN is beyond the capability of the topology model itself. Fortunately, PSNRP works on it, which will be further illustrated in Sect. 4.

4 Predictable Satellite Network Routing Protocol

In this section, PSNRP is illustrated in detail.

4.1 Some Definitions

It is necessary to define some concepts first of all.

LEO group and group manager (GM): a LEO group, denoted by $G(M_{i,j})$, refers to a set of LEO satellites all locating in the footprint of the same MEO satellite $M_{i,j}$ and constructing data links with it according to the principle of the longest covering time. $G(M_{i,j}) = \{L_{i,j,k} \mid k = 1, 2, \ldots, N_{i,j}\}$, in which $N_{i,j}$ means the number of LEO satellites in group $G(M_{i,j})$. $M_{i,j}$ is the group manager for all LEO satellites in group $G(M_{i,j})$, denoted by $GM(L_{i,j,k}) = M_{i,j}$.

Snapshot Sequence Report (SSR): a snapshot sequence report is a k-number topology snapshot array generated by ground Control Centers. It results from an off-line topology snapshot pre-calculation.

$$SSR(t_k) = <S_{k+1}, S_{k+2}, \ldots, S_{k+n}> \ , \ n \geq 0; \tag{1}$$

In which S_i refers to a topology snapshot. The equation shows that even before time t_k, S_{k+1} and the successful $n-1$ snapshots have been pre-calculated on the ground.

Snapshot Report (SR): A Snapshot Report is a topology snapshot handoff report. MEO satellite sends it to all its group members in LEO layer.

The report is constituted in the form of the differences between current snapshot and the next one, thus it is a difference report.

Link-state database (LSDB): Compared with conventional link-state protocols, the LSDB in PSNRP does not change a lot. After receiving the SR from group manager, a LEO satellite renews the topology and updates its LSDB with the graph extracted from SR.

Simple Link-State Advertisement (sLSA): When link anomalies are detected, a LEO satellite flood the sLSAs within LEO layer to inform other nodes. To reduce overhead, sLSA retains only four essential parameters: $sLSA(t) = <x, y, t, b>$, $(x, y) \in E_i$, $t_i \leq t \leq t_{i+1}$ $b = \{0, 1\}$; in which x, y are LEO satellites, t is the moment detecting an anomaly, b is a sign bit (0 refers to unplanned shut-down and 1 refers to unplanned turn-on of some link). We will see that due to the potentially unpredictable topology changes, the network may generate a lot of advertisements. In this situation, using sLSA greatly reduces the overhead.

LEO layer routing table: PSNRP provides two kinds of routing service—ordinary routing services and QoS routing services. Ordinary services aim to minimize the end-to-end delay of a call, which could meet the demand of most network traffic today. QoS services are prepared for users with QoS requirements such as delay, bandwidth, jitter, package loss, etc.

To emphasize the protocol itself, we do not consider QoS routing in this section, and leave them for Sect. 5. In PSNRP, generally speaking, all data packets are switched in LEO layer if possible, and MEO layer is just responsible for computing and distributing the control information. So routing tables are completed in LEO satellites, $LT(L_{i,j,k}) = \{<y, \mathrm{SPF}(L_{i,j,k}) \rightarrow y> \mid \forall y, y \text{ is a LEO satellite}\}$. In which SPF() represents the optimal path to every LEO satellite computed through Dijkstra algorithm.

Anomaly Report (AR): There are two kinds of AR: $\mathrm{AR}_{L \rightarrow M}$ and $\mathrm{AR}_{M \rightarrow G}$. When a LEO satellite detects anomalies, besides flooding sLSAs, it also reports to group manager through $\mathrm{AR}_{L \rightarrow M}$. Once the group manager receives the report, it uses the report to update snapshot S_{i+1} which is to be sent to LEO group members. By this mechanism, underlying sLSA flooding possibility is greatly reduced. If an anomaly lasts longer than the threshold, after receiving repeated ARs on it, the group manager sends $AR_{M \to G}$ to inform Control Centers of this long-term anomaly. The Control Centers then uses $AR_{M \to G}$ to update the computed snapshots.

Though the improved Virtual Topology Grouping strategy proposed by Long [15] has effectively reduced the total number of timeslots in a cycle, it is still infeasible to store all these snapshots on board.

Algorithm 1 describes steps taken by a LEO satellite node.

Algorithm 1: main loop for a LEO satellite $L_{i,j,k}$

```
1     var sr, x, slsa, ar /*sr is a SR variable; x is an anomaly; slsa is a sLSA variable;
                 ar is an AR variable*/
2   while true do
3       if sr = receive(IOL$_{M \to L}$) then /*receiving SR report from group manager*/
4           update_LSDB(sr);
5       end if
6       if x = detect_change() then /*detecting exception and storing it in x*/
7           update_LSDB(x);
8           slsa = create_sLSA(x); /*generating sLSA report*/
9           send (slsa, neighbor_nodes($L_{i,j,k}$)); /*flooding sLSA*/
10          ar = creat_AR(x);
11          send(ar, GM($L_{i,j,k}$)); /*sending AR report to group manager*/
12      end if
13      if slsa = receive(ISL) then /*receiving sLSA*/
14          if is_fresh(slsa) then
15              update_LSDB(slsa);
16              send (slsa, neighbor_nodes($L_{i,j,k}$))
17          end if
18      end if
19      update_LT(LSDB); /*updating routing table in LEO layer*/
20    end while
```

Packets are mainly routed in LEO layer. A satellite $L_{i,j,k}$ maintains a LSDB and updates its routing table according to this LSDB. The design for algorithms in LEO is mainly based on how to cope with two kinds of topology changes—predictable changes and unpredictable changes. Lines 3–5 describe the response to predictable changes, which are abstracted into a sequence of snapshots maintained by MEO satellites. When its time to handoff, group managers send new snapshot to all LEO group members through AR reports before the old snapshot expires. After receiving a SR report, a LEO satellite just need to update its LSDB according to the SR report. Lines 6–18 illustrate the mechanism for unpredictable changes. Unpredictable changes are often classified into two categories: the first happens at the initial stage of a new time slot, it is detected when a LEO satellite compares the updated LSDB with the real link-state; the other is that detected after the routing tables have been established. Whatever anomaly is detected, the node updates its LSDB first of all (Line 7) and then generates sLSA and AR (Lines 8–12), which are, respectively, to be sent to neighbor nodes and group managers. Similarly, if the node receives a sLSA, which has not been received yet, the node uses it to renew LSDB and keeps flooding. Given all the predictable and unpredictable changes, a node always calculates its routing table with the latest LSDB. Reference [16] proved that, for a link-state routing, optimal path re-calculation does not require a complete run of Dijkstra algorithm. Instead, nodes and links affected by SR or sLSA are sufficient. This is particularly beneficial in our model since a topology change usually affects few links or nodes. Our on-board computation ability could totally meet these computational requirements.

Algorithm 2: main loop for a MEO satellite $M_{i,j}$

```
var sr, ar /* sr is a SR variable; ar is an AR variable */
 while true do
    if timestamp(stack_next) == current_time()/*load back-up sequence*/
    then stack_now = stack_next;
            stack_next = 0;
    end if
    if timestamp(δ(stack_now)) == current_time()/* handoff time*/
    then sr = create_SR(δ(stack_now));
            sr = sr + $stack_{AR\_M}$ ;/*considering anomalies*/
            send (sr, $G(M_{i,j})$);
            stack_now = stack_now/δ(stack_now);
    end if
    if ssr = receive($UCL_{G \to M}$) then/*receiving SSR*/
        if isfresh(ssr) then
            stack_next= ssr;
            send(ssr, neighbor_nodes($M_{i,j}$));    /*flooding*/
        end if
    end if
    if ar = receive($IOL_{L \to M}$ ) then
        if isfresh(ar) then
            update_ $stack_{AR\_M}$ (ar);
            if isoverthreshold(ar) then /*detecting long-term anomalies*/
                send(create_$AR_{M \to G}$(ar), CC);
            end if
            send(ar, neighbor_nodes($M_{i,j}$));
        end if
    end if
    if ar = receive(ISL) then /*ordinary anomalies*/
        if isfresh(ar) then
            update_ $stack_{AR\_M}$ (ar);
            send(sr, neighbor_nodes($M_{i,j}$));
        end if
    else if ssr = receive(ISL) then /*receiving SSRfrom CCs*/
          if isfresh(ssr) then
              stack_next= ssr;
              send(ssr, neighbor_nodes($M_{i,j}$);
          end if
    end if
 end while
```

Algorithm 2 describes the main loop of algorithm in MEO nodes. MEO layer is the core of the whole PSNRP. Under normal circumstances, this layer does not participate in data routing or forwarding, but it is responsible for controlling all the space-based and ground-based sub-networks. A MEO satellite maintains three databases: stack_now is used to store the sequence whose snapshots are being downloaded to LEO group members one by one; stack_next stores the sequence which has been received from CCs, and the sequence will be sent to stack_now after stack_now being vacuumed up (Lines 3–6); $\mathrm{stack}_{\mathrm{AR\_M}}$ is used to store the anomalies reported by group members. Funtion $\delta()$ returns the first element in an array. When it is time to send a snapshot in stack_now to LEO layer for handoff, a MEO node sends a SR report on the basis of snapshot and latest anomalies in $\mathrm{stack}_{\mathrm{AR\_M}}$ (Line 9). Finally, stack_now eliminates the obsolete snapshot. On receiving a new SSR report from CCs or other nodes, a MEO node extract the sequence and store it in stack_next (Lines 13–18, Lines 32–36). Similarly, the MEO node updates its $\mathrm{stack}_{\mathrm{AR\_M}}$ with arriving AR reports. A very important function is isoverthreshold(), which returns the decision whether an anomaly is long-term or not by a comparison with a threshold. To reduce processing overhead, as illustrated above, long-term anomalies are reported to CCs by $\mathrm{AR}_{\mathrm{M}\rightarrow\mathrm{G}}$ to update SSR report.

PSNRP could effectively handle anomalies by regarding them as unpredictable changes, so it is robust on anomalies.

5 Simulation and Results

In this section, we mainly aim to show the performances of PSNRP, on which runs Dijkstra routing algorithm, by comparing it with SGRP. The simulation is based on the constellation model in Sect. 2, which is the same as that in [12]. Simulation tools are NS2 and OPNET. LEO layer provides accesses to MLSN for the users.

Figure 2 depicts the relationship between average end-to-end delays and their distances in three protocols. The results show that if the distances are short, their performances are similar; but with the increase of the distances PSNRP obviously has a shorter average end-to-end delay than SGRP. The reason is probably that when the distances exceed the radius of the group, most user data are routed through MEO satellites in SGRP—the propagation delays of IOLs are much longer than that of LEO layer ISLs.

We also test protocol’s performances under the circumstance of link congestions and satellite failures. PSNRP has a strong robustness on satellite failure and Fig. 3 depicts the result. The simulation studies the relationship between the failing probability of each satellite and the average convergence time. As shown by the figure, PSNRP performs far better than SGRP if the failing probability is not large. It may result from the essential sLSA broadcasting mechanism that makes all LEO nodes be aware of the topology change soon. Instead, in SGRP, after detecting the failure, a LEO node has to report to group manager and wait for the

Fig. 2 Average end-to-end delay

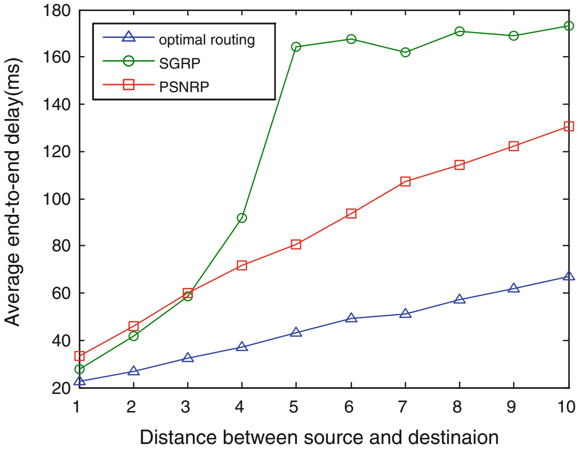
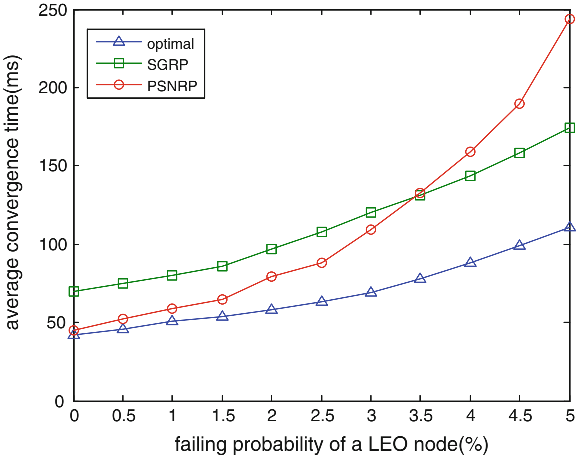


Fig. 3 Robustness on failures



re-calculated routing table. If the number of failed satellites is too large, too many sLSAs are transferred in the network for all the nodes to change LSDB, which causes huge overhead. SGRP is relatively insensitive to increment of the number, so it performs better. Considering that satellite failures are not common, PSNRP is more robust on satellite failures. The satellite failure in this paper only means LEO satellite failure. As the MEO satellites play central roles in both protocols, no failure is tolerable in MEO layer—even one would lead to the failure of the protocol. Actually, back-up MEO satellites are often deployed on MEO layer in case of failures.

6 Conclusion

In this paper, we propose a routing scheme for predictable satellite networks. Different from conventional MLSN routing protocols like SGRP, PSNRP takes full advantage of the network predictability to reduce on-board computation; also it introduces the ground Control Centers to utilize ground-based resources. The simulation results show that PSNRP obtains shorter end-to-end delays and stronger robustness on anomalies.

The future research will focus on how to solve the problem of inconsistence between the real topology and that maintained by nodes' LSDB at the initial stage of a snapshot. Another topic is how to reduce the overhead of protocol controlling data when a large number of anomalies happen within a short time.

Acknowledgments The work is jointly supported by the National Natural Science Foundation of China (Grants No. 60973145, 61004021).

References

1. Elbert BR (2004) The satellite communication application handbook. Artech House Inc., Norwood
2. Zhou Y, Sun F, Zhang B (2006) A time-division QoS routing protocol for three-layered satellite networks. Chin J Comput 29:1813–1822
3. Chang HS, Kim BW, Lee C (1998) FSA-based link assignment and routing in low-earth orbit satellite networks. IEEE Trans Veh Technol 47:1037–1048
4. Werner M, Delucchi C, Vegel H (1997) ATM-based routing in LEO/MEO satellite networks with intersatellite links. IEEE J Sel Areas Commun 15:69–81
5. Gounder V, Prakash R, Abu-Amara H (1999) Routing in LEO-based satellite networks. In: Proceedings of IEEE emerging technologies symposium on wireless communications and systems. IEEE Press, pp 91–96
6. Hashimoto Y (1998) Design of IP-based routing in a LEO satellite network. In: Proceedings of 3rd international workshop on satellite-based information services, Dallas, pp 81–88
7. Ekici E, Akyildiz IF, Bender MD (2000) Datagram routing algorithm for LEO satellite networks. In: Proceedings of IEEE INFOCOM 2000. IEEE Press, pp 500–508
8. Zhu L, Wang R (2004) The research of satellite routing algorithm. Microcomput Dev 14:9–11
9. Reeves CR (ed) (1993) Modern heuristic techniques for combitiorial problems. Blackwell Scientific Publications, Oxford
10. Fischer D, Basin D, Engel T (2008) Topology dynamics and routing for predictable mobile networks. In: Proceedings of ICNP 2008. IEEE Communications Society, Orlando, pp 207–217
11. Ekici E, Akyildiz IF, Bender MD (2002) MLSR: a novel routing algorithm for multi-layered satellite IP networks. IEEE Trans Networking 18:411–424
12. Chen C, Ekici E, Akyildiz IF (2002) Satellite grouping and routing protocol for LEO/MEO satellite IP networks. In: Proceedings of 5th ACM international workshop on wireless mobile multimedia (WoWMoM'2002). pp 109–116
13. Long F (2010) Research on satellite network robust QoS-aware routing, PhD thesis. Tsinghua University, Beijing

14. North J (1994) The Fontana history of astronomy and cosmology. Fontana Press, London, pp 309–310
15. Long F, Xiong N, Athanasios VV, Yang LT, Sun F (2009) A sustainable heuristic QoS routing algorithm for pervasive multi-layered satellite wireless networks. Wireless Netw 14:1–17
16. OSPF protocol analysis, RFC (July 1991) Internet Engineering Task Force (IETF) Request for Comments 1245