Simple Change Adaptive Routing Algorithm for Satellite IP Networks

Liu Feng, Zhang Yu
School of Computer and Information Science, Southwest University, Chongqing, 400715, China
Email :{ liuf, zhangyu}@swu.edu.cn

***Abstract*— although the LEO constellation has dynamic satellite network topology, the topology changes are predictable and periodic. Based on the characteristics above, a new simple change adaptive routing algorithm (SCARA) is proposed for satellite IP networks. SCARA uses simple rules to decide the next hop. The rules are 1) using only two possible directions to reduce the routing path; 2) assigning higher priority to the direction with higher satellite latitude; 3) avoiding the polar region;4) avoiding congestion with local buffer and queue information. SCARA is a kind of distributed routing algorithm. Only local information, such as satellite latitude, logical location, buffer, source and destination address, etc., is needed to route packets, and information from other satellites is not necessary for routing. This feature makes SCARA can be used in satellite IP networks. A simulator is developed for evaluating the performance. The simulation result shows that SCARA has the capability to find the minimum path, lower end-to-end delay and lower time complexity.**

***Index Terms*—routing algorithm; satellite IP networks; LEO constellation; local information; simulation**

I. INTRODUCTION

Satellite networks have the capabilities to support wide range of data communication and Internet applications. It exhibits unique features and offers an array of advantages over traditional terrestrial networks[1]. They are an attractive option to provide broadband integrated Internet services to globally scattered users, due to their potential advantages such as extensive geographic coverage, high bandwidth availability, and inherent broadcast capabilities[2]. Definitely, satellite networks will become an integral part of Internet.

Low Earth Orbit (LEO) satellites have some distinct characteristics such as low propagation delay, low power requirements and more efficient spectrum allocation[3]. However, they move at rapid speeds relative to the ground terminals[4]. The high mobility and changing network topology make terrestrial Internet routing protocols, such as OSPF[5] and RIP[6], no longer effective when used in LEO satellite networks. These protocols rely on exchanging topology information when network connections are established or changed. In LEO constellations, this topology information quickly becomes obsolete and must constantly be refreshed with new information. The overhead of regularly providing this information is an obstacle to considering satellites as conventional Internet routers[4]. So the new routing schemes should be designed for LEO satellites networks.

Previous routing schemes[7-10] for ATM or ATM-type switches onboard satellites are designed based on the connection-oriented mechanisms that satellite networks own[11]. However, with the increasing of Internet based traffic, there is huge demand for satellite networks to provide connectionless communication service. Research work is carried out in [4, 12-20] to find routing scheme for connectionless satellite networks.

Routing protocols based on dynamic topology are introduced by [12, 13]. In [12], A LEO satellite network is modeled as a finite state automaton, where the system period is divided into states. It is proposed to perform optimal routing on each of these fixed topologies for the best use of ISL in the system. The virtual node concept is introduced in [4] to exploit the regularity of the constellation's topology. Routing is performed in the fixed virtual topology by the use of a common routing protocol. This scheme can directly integrate the space network with the terrestrial IP network and may provide good support for IP-multicast and IP-QoS[4]. [13] formalizes the construction of fault blocks by a state transition model based on Finite State automaton. X-Y Boundary Routing Algorithm (X-YBRA) was presented for deadlock-free fault-tolerant adaptive routing outside the fault blocks. X-YBRA simulation shows its diffusion overhead is lower than traditional routing algorithms such as Distance Vector and Link State Routing algorithm. However, [21] points out that this approach presents some challenging problems such as scalability of routing tables and high computational complexity in space devices. One-to-one mapping of physical topology to a virtual topology is also problematic for several reasons[17].

MLSR[14], SGRP[15], ELB[16] etc. presented multi layered routing protocols. Multi layered architectures are used to improve the performance of satellite IP networks. MLSR introduced a multilayered satellite IP network consists of LEO, medium-Earth orbit (MEO), and geostationary Earth orbit (GEO) satellites, and developed a multilayered satellite routing algorithm which calculates routing tables efficiently using the collected delay measurements. MLSR provides higher performance than the satellite networks with fewer satellite layers when the network load is high. SGRP[15] is a multi-layered satellite network routing protocol based on snapshots. It divides Low Earth Orbit (LEO) satellites

doi:10.4304/jsw.8.8.1991-1999

into groups according to the footprint area of the MEO satellites in each snapshot period. Based on the delay reports sent by LEO satellites, MEO satellite managers compute the minimum delay paths for their LEO members. SGRP gets better performance than Datagram Routing Algorithm[18] in the simulation. ELB[16] proposed Explicit Load Balancing scheme for using multi layer satellites for routing in LEO networks which experience heavy traffic. The range for exchanging traffic load information is extended to achieve reductions in packet drop rates. These routing algorithms normally get higher performance than one layer satellites networks. However, multi layer architecture needs satellites acting as routers running in higher layers, which increase the complex and expense of the satellite system.

Snapshots based routing is used in [15] [20]. It is proved to be an effective solution for LEO satellite constellation networks. However, the routing algorithm is computationally complex with the need to calculate satellite positions and the shortest paths. Moreover, these predetermined routings can neither adapt to unbalanced dynamical traffic loads nor provide QoS guarantees[22].

[17-19] introduced routing protocols trying to minimize the routing path. [19] presents a modification of the “minimization of number of hops” algorithm which allows to reduce the total path length, and the end-to-end delay. Distributed algorithms are presented in [17] for minimizing the propagation delay. Depending on the geographic information embedded in the addresses, each satellite forwards the packet to its neighbor that most reduces the distance to the destination. [18] introduced the datagram routing algorithm (DRA) for an idealized polar constellation. It regards the satellite network as a mesh topology consisting of logical locations (virtual nodes). Data packets are routed in a distributed fashion in this fixed topology. DRA consists of two phases: at a given satellite hop, it first finds all the neighboring satellites that can move the packet one hop closer to the destination. Then, from the candidate next hops, it selects the one that most reduces the remaining distance to the destination.

After looking into the characteristics of low earth orbit satellite constellation, we introduce a new simple algorithm (SCARA) for LEO satellites routing. The algorithm is based on the LEO constellation features[21], such as dynamic satellite network topologies, predictable and periodic network topology changes, fixed number of nodes, non-homogeneous and dynamic traffic, etc. SCARA only uses the local information, such as satellite latitude, logical location, output buffer used, source and destination address, etc., for routing packets, and does not need the information from other satellite. This makes SCARA can forward IP packets through LEO satellite constellation. SCARA tries to find the shortest path for packets forwarding. Four simple rules are used when deciding the next hop. 1) There are two directions help to reduce the routing path; 2) the direction with higher satellite latitude has the priority; 3) try to avoid entering the polar region; 4) local buffer and queue information is used to avoid congestion.

The paper is organized as follows. In Section II, the LEO satellite network architecture is presented. The metrics used in the paper are given in Section III. In Section IV, the new routing algorithm, SCARA, is presented in detail. The performance evaluation of SCARA is presented in Section V. Finally, Section VI concludes the paper.

II LEO SATELLITE NETWORK ARCHITECTURE

The satellite network is composed of N_p planes, each with satellites at low distances from the Earth. The planes are separated from each other with the same angular distance of $180^\circ / N_p$. They cross each other only over the North and South poles. There are N_s satellites in each plane. The satellites in a plane are separated from each other with an angular distance of $360^\circ / N_s$. Since the planes are circular, the radii of the satellites in the same plane are the same at all times and so are the distances from each other[18].

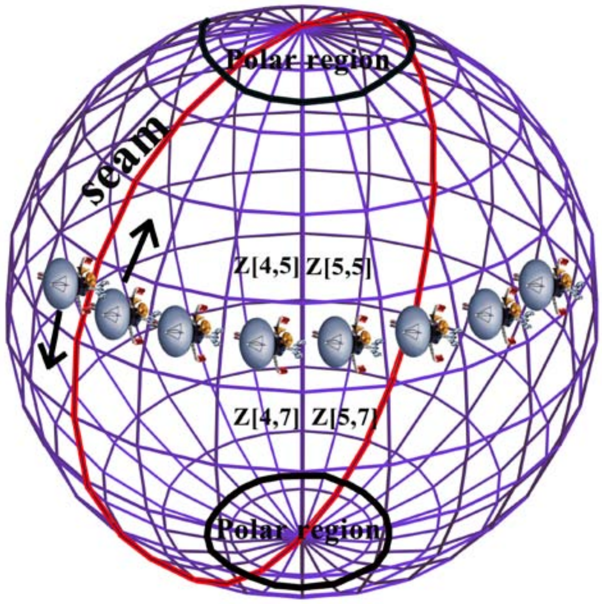

Figure 1. LEO satellite network architecture

Each satellite is identified by S_{ij}, where i is the plane number, $0 \le i \le N_p - 1$, and j is the satellite position in the plane, $0 \le j \le N_s - 1$. The S_{ij} represents the relative position of a satellite. Its value is assigned when LEO satellite constellation is constructed and does not change with time. If S_{ij} is the source or sink of a packet, it can also be referred as S_{src} or S_{dst}, which represents the source satellite and destination satellite respectively. Both geographical location and logical location are used in the architecture. The geographical location holds the value of longitude and latitude of satellite. The latitude of S_{ij} is referred to as $Lat(S_{ij})$. The logical location concept is used for the LEO layer to hide the mobility of LEO satellites. Logical locations are equally spaced points in the grid of the LEO satellite constellation and are embodied by the nearest LEO satellites. A logical location is a zone mapped to the earth as shown in Fig.1, and is referred to as Z_{ij} , $0 \le i \le N_p - 1$ and $0 \le j \le N_s - 1$. $Z_i(S_{ij})$ and $Z_j(S_{ij})$ identify the zone in which satellite S_{ij} locates currently. The logical location does not move and are filled by the nearest satellite. Hence, the identity of the satellite is not permanently coupled with its logical location, which is taken over by the successor satellite in the same plane[18]. Each satellite has four neighboring satellites: two in the same plane and two in the left and right planes. The links between satellites in the same plane are called intraplane ISLs. Intraplane ISL from satellite S_{ij} to S_{ik} is referred as $nISL(S_{ij} \to S_{ik})$. The links between satellites in different planes are called interplane ISLs. Interplane ISL from satellite S_{ik} to S_{jk} is referred as $wISL(S_{ik} \to S_{jk})$. Each satellite is able to set up four ISLs connecting to its four immediately neigh-boring satellites. On intraplane and interplane ISLs, the communication is bidirectional. The intraplane ISLs are maintained at all times, i.e., each satellite is always connected to the rest of the network through its up and down neighbors[18]. Interplane ISLs can be temporarily switched-off when the viewing angle or distance between two satellites changes too fast for the steerable antennas to follow. This situation may occur between two counter-rotating orbits or when two planes approach or cross.

When routing a packet, a satellite can find zero, one or several ISLs from its ISLs available. The first direction, referred as Dir_{1st} , has the highest priority to route the packet. Satellites also can pick up the second, third and fourth direction as backup route, which are referred as Dir_{2nd} , Dir_{3rd} and Dir_{4th} respectively. If there is none ISL for routing, then the packet will have no movement and stay in the satellite S_{src} .So Dir_x may have a value from five defined in (1), where $x = Dir_{1st}, .., Dir_{4th}$.

$$Dir_x = \begin{cases} ISL_{none} : no\ movement \\ ISL_{up} : through\ upward\ ISL \\ ISL_{down} : through\ downward\ ISL \\ ISL_{right} : through\ right\ ISL \\ ISL_{left} : through\ left\ ISL \end{cases} \quad (1)$$

All satellites are moving in the same circular direction within the same plane. As a consequence, any satellite that is observed from the Earth moving from south to north will be observed to start moving from North to South when it crosses the North Pole. Hence, the 0th and $(N_p - 1)$ th planes rotate in opposite directions. The borders of counter-rotating satellites are called seams[18], as shown in Fig.1.Seam divides the network into two parts and the satellites over the eastern hemisphere and the satellites over the western hemisphere move in opposite directions. In eastern hemisphere satellites move from south to north, while in western hemisphere they move from north to south. It is assumed that there is has interplane ISLs passing the seam. Hence, a data that originates from a location at one hemisphere could be sent to a location in the other hemisphere without passing a polar.

III METRICS

A. End-to-end Delay

When designing the routing algorithm, a suitable link cost metric must be introduced firstly, which can represent the information of network status[23]. End-to-end delay is the most important metric used in this paper. End-to-end delay is the sum of propagation delay, processing delay and queuing delay. All satellites are assumed to have the same processing capability. So the processing delay equals each other and keeps constant and does not be calculated for simplicity. Propagation and queuing delay are two dynamically changing parameters, which have significant effects on the routing performance. The propagation delay is determined by ISLs' distances. All intraplane ISLs' propagation delay is same and don't change and can be calculated in advance. While interplane ISLs' propagation delay are different and change with the satellites' location. The queuing delay is affected by the traffic load on a particular satellite and its outgoing links, as the satellite's coverage traverses varying traffic zones.

The delay, which is composed of propagation, processing, queuing, and transmission delays, can vary greatly with the changes of the positions of the individual satellites and the network load[24].

B. Propagation Delay

The length of intraplane ISL from S_{ia} to S_{ib} is referred as $D_{ia\to ib}$. It is computed by

$$D_{ia\to ib} = \sqrt{2}R\sqrt{1-\cos\left(360^\circ / N_s\right)},$$

where R is the radius of the plane.

The length of interplane ISL $D_{aj\to bj}$ is variable and is calculated by

$$D_{aj\to bj} = \sqrt{2}R\sqrt{1-\cos\left(180^\circ / N_p\right)} \times \cos(Lat(Z_{aj})),$$

where $Lat(Z_{aj})$ is the latitude of Z_{aj} .

The propagation delay of intraplane and interplane ISLs can be calculated by $D_{ia\to ib} / c$ and $D_{aj\to bj} / c$ respectively, where c is the velocity of light. Because the length between intraplane ISLs is same, the propagation delay on the intraplane links is always fixed. While propagation delay on the interplane links is changing, for the length between interplane ISLs is changing.

C. Queue Delay

The user density level and host density level of zone Z_{ab} are defined as U_{ab} and H_{ab} . The inter-satellite traffic requirement $T_{ab\to ij}$ between zone Z_{ab} and Z_{ij} depends on the U_{ab} , the H_{ab} , and the distance

$D_{ab\to ij}$ between the zones. It can be calculated according to formula(2).

$$T_{ab\to ij} = \frac{(U_{ab}H_{ab})^{\alpha}}{(D_{ab\to ij})^{\beta}} \quad (2)$$

In the simulations, we set α = 0.5 and β = 1.5 (same as defined in [24]]). The average packet arrival rate is computed by formula(3).

$$\lambda_{ab\to ij} = \frac{T_{ab\to ij}}{\sum_{\forall cd}\sum_{\forall xy} T_{cd\to xy}} \times T_{offered} \quad (3)$$

Where $T_{offered}$ represents total offered traffic generated worldwide.

In a LEO satellite network, an inter-satellite link can be modeled as a finite capacity queue[23]. The arrival is assumed to follow Poisson process with rate λ. Service time of satellite S_{ij} may be exponential distributed with service rate $\mu_{S_{ij}}$.

IV SIMPLE CHANGE ADAPTIVE ROUTING ALGORITHM

A Direction Estimation

SCARA is a distributed routing algorithm. It does not need other satellites' information when deciding the next hop. When a packet reached a satellite, the satellite checks the packet's destination. If the destination equals the zone of the satellite, then the satellite consumes the packet or forwards the packet to the gateway on earth. If not, the satellite will find a direction and forward the packet through it. Hence, the role of the satellite is the source of the packet and is marked as S_{src} .

When estimating the direction, there are four possible choices. They are Dir_{1st} , Dir_{2nd} , Dir_{3rd} and Dir_{4th} , which take priorities from the highest to the lowest. When SCARA finished, it will assign values to them. There are three situations for SCARA to consider respectively when estimating the directions. The first situation is that both S_{src} and S_{dst} locates in a same seam. The second situation is that S_{src} locate in west hemisphere while S_{dst} locate in east hemisphere. The third situation is that S_{src} locate in east hemisphere while S_{dst} locate in west hemisphere.

a) Both S_{src} and S_{dst} locate in a same seam

When S_{src} and S_{dst} locate in a same seam, packets from S_{src} should be routed to S_{dst} directly through vertical or horizontal ISL, without passing the seam or crossing a polar. The routing rules of SCARA are defined as follows.

1) $Z_i(S_{src}) < Z_i(S_{dst})$

If $Z_i(S_{src}) < Z_i(S_{dst})$, a necessary operation is forwarding the packet through the right ISL of S_{src}. The information of $Z_j(S_{src})$ and $Z_j(S_{dst})$ is needed to decide whether ISL_{right} shall be the first or the second direction.

- $Z_j(S_{src}) < Z_j(S_{dst})$

If $Lat(S_{src}) \ge Lat(S_{dst})$, then Dir_{1st} is set to ISL_{right} and Dir_{2nd} is set to ISL_{down};

If $Lat(S_{src}) < Lat(S_{dst})$, then Dir_{1st} is set to ISL_{down} and Dir_{2nd} is set to ISL_{right}.

- $Z_j(S_{src}) > Z_j(S_{dst})$

If $Lat(S_{src}) \ge Lat(S_{dst})$, then Dir_{1st} is set to ISL_{right} and Dir_{2nd} is set to ISL_{up};

If $Lat(S_{src}) < Lat(S_{dst})$, then Dir_{1st} is set to ISL_{up} and Dir_{2nd} is set to ISL_{right}.

- $Z_j(S_{src}) = Z_j(S_{dst})$

Dir_{1st} is set to ISL_{right} and Dir_{2nd} is set to ISL_{none}.

2) $Z_i(S_{src}) > Z_i(S_{dst})$

If $Z_i(S_{src}) > Z_i(S_{dst})$ a necessary operation is forwarding the packet through the left ISL of S_{src}. The information of $Z_j(S_{src})$ and $Z_j(S_{dst})$ is needed to decide whether ISL_{left} shall be the first or the second direction.

- $Z_j(S_{src}) < Z_j(S_{dst})$

If $Lat(S_{src}) \ge Lat(S_{dst})$, then Dir_{1st} is set to ISL_{left} and Dir_{2nd} is set to ISL_{down};

If $Lat(S_{src}) < Lat(S_{dst})$, then Dir_{1st} is set to ISL_{down} and Dir_{2nd} is set to ISL_{left}.

- $Z_j(S_{src}) > Z_j(S_{dst})$

If $Lat(S_{src}) \ge Lat(S_{dst})$, then Dir_{1st} is set to ISL_{left} and Dir_{2nd} is set to ISL_{up};

If $Lat(S_{src}) < Lat(S_{dst})$, then Dir_{1st} is set to ISL_{up} and Dir_{2nd} is set to ISL_{left}.

- $Z_j(S_{src}) = Z_j(S_{dst})$

Dir_{1st} is set to ISL_{left} and Dir_{2nd} is set to ISL_{none}.

3) $Z_i(S_{src}) = Z_i(S_{dst})$

If $Z_i(S_{src}) = Z_i(S_{dst})$ there shall be no horizontal movement. The vertical direction is estimated with information from $Z_j(S_{src})$ and $Z_j(S_{dst})$.

If $Z_j(S_{src}) = Z_j(S_{dst})$, then Dir_{1st} is set to ISL_{down} and Dir_{2nd} is set to ISL_{none};

If $Z_j(S_{src}) > Z_j(S_{dst})$, then Dir_{1st} is set to ISL_{up} and Dir_{2nd} is set to ISL_{none}.

If $Z_j(S_{src}) = Z_j(S_{dst})$, then Dir_{1st} is set to ISL_{none} and Dir_{2nd} is set to ISL_{none}.

b) S_{src} locates in west hemisphere and S_{dst} locates in east hemisphere

When S_{src} locates in west hemisphere and S_{dst} locates in east hemisphere, minimal horizontal hops through left ISL $MinHops_L$ can be calculated by $MinHops_L = Z_i(S_{src}) + 1 + N_p - Z_i(S_{dst})$; minimal horizontal hops through right ISL $MinHops_R$ can be calculated by $MinHops_R = N_p - Z_i(S_{src}) + Z_i(S_{dst}) + 1$.

1) $MinHops_L \geq MinHops_R$

- $(N_s - 1 - Z_j(S_{src})) < Z_j(S_{dst})$

If $Lat(S_{src}) \geq Lat(S_{dst})$, then Dir_{1st} is set to ISL_{right} and Dir_{2nd} is set to ISL_{up};

If $Lat(S_{src}) < Lat(S_{dst})$, then Dir_{1st} is set to ISL_{up} and Dir_{2nd} is set to ISL_{right}.

- $(N_s - 1 - Z_j(S_{src})) > Z_j(S_{dst})$

If $Lat(S_{src}) \geq Lat(S_{dst})$, then Dir_{1st} is set to ISL_{right} and Dir_{2nd} is set to ISL_{up};

If $Lat(S_{src}) < Lat(S_{dst})$, then Dir_{1st} is set to ISL_{up} and Dir_{2nd} is set to ISL_{right}.

- $(N_s - 1 - Z_j(S_{src})) = Z_j(S_{dst})$

Dir_{1st} is set to ISL_{right} and Dir_{2nd} is set to ISL_{none}.

2) $MinHops_L < MinHops_R$

- $(N_s - 1 - Z_j(S_{src})) < Z_j(S_{dst})$

If $Lat(S_{src}) \geq Lat(S_{dst})$, then Dir_{1st} is set to ISL_{left} and Dir_{2nd} is set to ISL_{up};

If $Lat(S_{src}) < Lat(S_{dst})$, then Dir_{1st} is set to ISL_{up} and Dir_{2nd} is set to ISL_{left}.

- $(N_s - 1 - Z_j(S_{src})) > Z_j(S_{dst})$

If $Lat(S_{src}) \geq Lat(S_{dst})$, then Dir_{1st} is set to ISL_{left} and Dir_{2nd} is set to ISL_{down};

If $Lat(S_{src}) < Lat(S_{dst})$, then Dir_{1st} is set to ISL_{down} and Dir_{2nd} is set to ISL_{left}.

- $(N_s - 1 - Z_j(S_{src})) = Z_j(S_{dst})$

Dir_{1st} is set to ISL_{left} and Dir_{2nd} is set to ISL_{none}.

c) S_{src} locates in east hemisphere and S_{dst} locates in west hemisphere

When S_{src} locates in east hemisphere and S_{dst} locates in west hemisphere, minimal horizontal hops through left ISL $MinHops_L$ can be calculated by $MinHops_L = Z_i(S_{src}) + 1 + N_p - Z_i(S_{dst})$; minimal horizontal hops through right ISL $MinHops_R$ can be calculated by $MinHops_R = N_p - Z_i(S_{src}) + Z_i(S_{dst}) + 1$.

1) $MinHops_L > MinHops_R$

- $(N_s - 1 - Z_j(S_{src})) < Z_j(S_{dst})$

If $Lat(S_{src}) \geq Lat(S_{dst})$, then Dir_{1st} is set to ISL_{right} and Dir_{2nd} is set to ISL_{up};

If $Lat(S_{src}) < Lat(S_{dst})$, then Dir_{1st} is set to ISL_{up} and Dir_{2nd} is set to ISL_{right}.

- $(N_s - 1 - Z_j(S_{src})) > Z_j(S_{dst})$

If $Lat(S_{src}) >= Lat(S_{dst})$, then Dir_{1st} is set to ISL_{right} and Dir_{2nd} is set to ISL_{down};

If $Lat(S_{src}) < Lat(S_{dst})$, then Dir_{1st} is set to ISL_{down} and Dir_{2nd} is set to ISL_{right}.

- $(N_s - 1 - Z_j(S_{src})) = Z_j(S_{dst})$

Dir_{1st} is set to ISL_{right} and Dir_{2nd} is set to ISL_{none}.

2) $MinHops_L < MinHops_R$

- $(N_s - 1 - Z_j(S_{src})) < Z_j(S_{dst})$

If $Lat(S_{src}) \geq Lat(S_{dst})$, then Dir_{1st} is set to ISL_{left} and Dir_{2nd} is set to ISL_{up};

If $Lat(S_{src}) < Lat(S_{dst})$, then Dir_{1st} is set to ISL_{up} and Dir_{2nd} is set to ISL_{left}.

- $(N_s - 1 - Z_j(S_{src})) > Z_j(S_{dst})$

If $Lat(S_{src}) \geq Lat(S_{dst})$, then Dir_{1st} is set to ISL_{left} and Dir_{2nd} is set to ISL_{down};

If $Lat(S_{src}) < Lat(S_{dst})$, then Dir_{1st} is set to ISL_{down} and Dir_{2nd} is set to ISL_{left}.

- $(N_s - 1 - Z_j(S_{src})) = Z_j(S_{dst})$

Dir_{1st} is set to ISL_{left} and Dir_{2nd} is set to ISL_{none}.

B. Direction Correction

When satellite enters polar region, its interplane ISL will be closed. So the direction estimated before have to be corrected. Following rules are used to recalculate for the first and second directions.

When S_{src} locates in the polar region, if ($Dir_{1st} = ISL_{left}$) or ($Dir_{1st} = ISL_{right}$) and if ($Dir_{2nd} = ISL_{up}$) or ($Dir_{2nd} = ISL_{down}$), then switch Dir_{1st} and Dir_{2nd}.

C. Congestion Avoidance

Final ISL direction is calculated according to the information of Dir_{1st}, Dir_{2nd} and the number of packets waiting in the queue. $TraDelay$ is the time needed to transmit a packet.

a) $Dir_{1st} = ISL_{left}$

If ($Dir_{2nd} = ISL_{left}$), then $Dir_{final} = ISL_{left}$;

If $(Dir_{2nd} = ISL_{right})$ and $(IslCount_{left}(S_{src}) \times TraDelay > IslCount_{right}(S_{src}) \times TraDelay + 2 \times 6.039)$, then $Dir_{final} = ISL_{right}$;

If $(Dir_{2nd} = ISL_{up})$ and $(IslCount_{left}(S_{src}) \times TraDelay > IslCount_{up}(S_{src}) \times TraDelay + 2 \times 6.039)$, then $Dir_{final} = ISL_{up}$;

If $(Dir_{2nd} = ISL_{down})$ and $(IslCount_{left}(S_{src}) \times TraDelay > IslCount_{down}(S_{src}) \times TraDelay + 2 \times 6.039)$, then $Dir_{final} = ISL_{down}$.

b) $Dir_{1st} = ISL_{right}$

If $(Dir_{2nd} = ISL_{left})$ and $(IslCount_{right}(S_{src}) \times TraDelay > IslCount_{left}(S_{src}) \times TraDelay + 2 \times 6.039)$,

then $Dir_{final} = ISL_{left}$;

If $(Dir_{2nd} = ISL_{right})$ then $Dir_{final} = ISL_{right}$;

If $(Dir_{2nd} = ISL_{up})$ and $(IslCount_{right}(S_{src}) \times TraDelay > IslCount_{up}(S_{src}) \times TraDelay + 2 \times 6.039)$,

then $Dir_{final} = ISL_{up}$;

If $(Dir_{2nd} = ISL_{down})$ and $(IslCount_{right}(S_{src}) \times TraDelay > IslCount_{down}(S_{src}) \times TraDelay + 2 \times 6.039)$,

then $Dir_{final} = ISL_{down}$.

c) $Dir_{1st} = ISL_{up}$

If $(Dir_{2nd} = ISL_{left})$ and $(IslCount_{up}(S_{src}) \times TraDelay > IslCount_{left}(S_{src}) \times TraDelay + 2 \times 6.039)$,

then $Dir_{final} = ISL_{left}$;

If $(Dir_{2nd} = ISL_{right})$ and $(IslCount_{up}(S_{src}) \times TraDelay > IslCount_{up}(S_{src}) \times TraDelay + 2 \times 6.039)$,

then $Dir_{final} = ISL_{right}$;

If $(Dir_{2nd} = ISL_{up})$ then $Dir_{final} = ISL_{up}$;

If $(Dir_{2nd} = ISL_{down})$ and $(IslCount_{up}(S_{src}) \times TraDelay > IslCount_{down}(S_{src}) \times TraDelay + 2 \times 6.039)$,

then $Dir_{final} = ISL_{down}$.

d) $Dir_{1st} = ISL_{down}$

If $(Dir_{2nd} = ISL_{left})$ and $(IslCount_{down}(S_{src}) \times TraDelay > IslCount_{left}(S_{src}) \times TraDelay + 2 \times 6.039)$,

then $Dir_{final} = ISL_{left}$;

If $(Dir_{2nd} = ISL_{right})$ and $(IslCount_{down}(S_{src}) \times TraDelay > IslCount_{up}(S_{src}) \times TraDelay + 2 \times 6.039)$,

then $Dir_{final} = ISL_{right}$;

If $(Dir_{2nd} = ISL_{up})$ and $(IslCount_{down}(S_{src}) \times TraDelay > IslCount_{up}(S_{src}) \times TraDelay + 2 \times 6.039)$,

then $Dir_{final} = ISL_{up}$;

If $(Dir_{2nd} = ISL_{down})$, then $Dir_{final} = ISL_{down}$.

V PERFORMANCE OF ROUTING ALGORITHM

A. Simulation Setup

We developed our simulator with Delphi. For each simulated routing protocol, the simulator outputs the corresponding end-to-end delay metric. Parameters used in the simulator are shown in TABLE. 1.

Average packet size is assumed to be 1000 bytes. So the link capacity becomes $C_{ij} = 20000$ packets per second and the buffer size becomes 5000 packets. The total traffic can be calculated according to formula(2):

$$\sum_{\forall cd}\sum_{\forall xy} T_{cd \to xy} = 0.00919591070476073.$$

TABLE I.
PARAMETERS OF LEO SATELLITE CONSTELLATION

Name	Value
Altitude	780 km
Number of plane	12
Number of satellite per plane	24
Angular velocity	3.6°/min
Orbit inclination angle	90
Number of intra-plane ISLs	2
Number of inter-plane ISLs	2
UDL capacity	160 Mbps
ISL capacity	160 Mbps
Outgoing link buffer size	5 MB
Polar regions latitudes	75.0°~ 90.0°

The total link capacity is

$$\sum_{\forall i}\sum_{\forall j} C_{ij} = N_p * N_s * C_{ij} = 12*24*20000.$$

Three types of routing protocols are evaluated using our simulator: SCARA, the Datagram Routing Algorithm [18], and the Dijkstra algorithm[25]. SCARA and DRA have the capabilities to adapt the change of link usage. Dijkstra algorithm will find the cost-lowest path based on the path distance[26]. We provide the same congestion avoidance ability as DRA has. It is referred as DWCA (Dijkstra with congestion avoidance) later for convenience.

Our experiments are based on the observation of the shortest path and end-to-end delay between certain terrestrial source-destination pairs. They are

$Pair1: Z[3,6] \to Z[7,11]$,

$Pair2: Z[2,23] \to Z[11,5]$,

$Pair3: Z[3,8] \to Z[2,18]$,

$Pair4: Z[7,23] \to Z[0,8]$,

$Pair5: Z[6,10] \to Z[8,19]$, and

$Pair6: Z[7,6] \to Z[8,20]$.

B. End-to-end Delay

Six source-destination pairs communication were simulated under $2\%, 4\%, 6\%, 8\%, 10\%, 12\%, 14\%$ and 16% link usage respectively. The delay metric is sampled every 0.5 second. Then the average end-to-end delay is calculated from the simulation results, as shown in Fig.2.

From the data in the figure, it can be known that SCARA has better performance than DRA and DWCA. $Z[3,6]$ and $Z[7,11]$ locate in the same seam, so the delay is lower than other pairs obviously. Packets form each pair routed by SCARA have lower average end-to-end delay. Because SCARA has the capabilities of finding the shortest path and can adapt the change of link congestion.

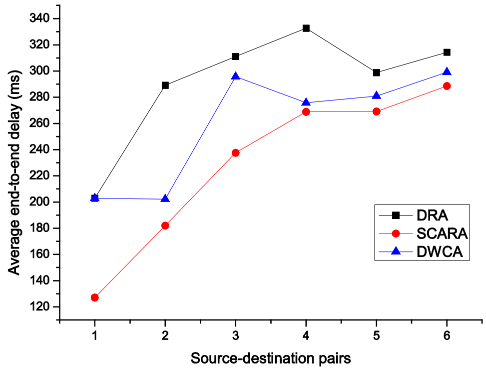

Figure 2. Average end-to-end delays

It can be known from the figure, for pair 1, DRA and DWCA give the same average end-to-end delay. However, for the rest of other pairs, DWCA performs better than DRA.

C. Minimum Path

We also checked the capabilities of finding the minimum delay path. All packets routed are saved and the shortest delay paths are picked up. Fig.3 shows the minimum path delay of each source-destination pair in DRA, SCARA and DWCA simulation.

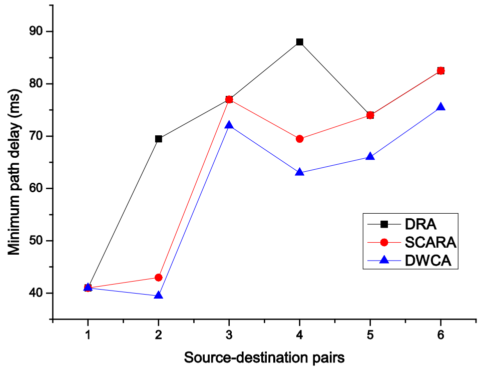

Figure 3. Minimum path delays

The paths found by DRA,SCARA and DWCA is as follows.

- For pair 1:
  DRA, SCARA and DWCA output the same path:
  Z[3,6]→Z[3,7]→Z[3,8]→Z[3,9]→Z[3,10]→Z[3,11]→Z[4,11]→Z[5,11]→Z[6,11]→Z[7,11].
- For pair 2:
  DRA:
  S[2,23]→S[2,0]→S[2,1]→S[3,1]→S[4,1]→S[5,1]→S[6,1]→S[7,1]→S[8,1]→S[9,1]→S[10,1]→S[11,1]→S[11,2]→S[11,3]→S[11,4]→S[11,5].
  SCARA:
  S[2,23]→S[2,22]→S[1,22]→S[0,22]→S[11,1]→S[11,2]→S[11,3]→S[11,4]→S[11,5].
  DWCA: :
  S[2,23]→S[1,23]→S[0,23]→S[11,0]→S[11,1]→S[11,2]→S[11,3]→S[11,4]→S[11,5].
- For pair 3:
  DRA and SCARA:
  Z[3,8]→Z[4,8]→Z[5,8]→Z[6,8]→Z[7,8]→Z[8,8]→Z[9,8]→Z[10,8]→Z[11,8]→Z[0,15]→Z[1,15]→Z[2,15]→Z[2,16]→Z[2,17]→Z[2,18].
  DWCA:
  Z[3,8]→Z[3,9]→Z[3,10]→Z[3,11]→Z[3,12]→Z[2,12]→Z[2,13]→Z[2,14]→Z[2,15]→Z[2,16]→Z[2,17]→Z[2,18]
- For pair 4:
  DRA:
  Z[7,23]→Z[7,0]→Z[7,1]→Z[6,1]→Z[5,1]→Z[4,1]→Z[3,1]→Z[2,1]→Z[1,1]→Z[0,1]→Z[0,2]→Z[0,3]→Z[0,4]→Z[0,5]→Z[0,6]→Z[0,7]→Z[0,8].
  SCARA:
  Z[7,23]→Z[7,22]→Z[8,22]→Z[9,22]→Z[10,22]→Z[11,22]→Z[0,1]→Z[0,2]→Z[0,3]→Z[0,4]→Z[0,5]→Z[0,6]→Z[0,7]→Z[0,8].
  DWCA:
  Z[7,23]→Z[8,23]→Z[9,23]→Z[10,23]→Z[11,23]→Z[0,0]→Z[0,1]→Z[0,2]→Z[0,3]→Z[0,4]→Z[0,5]→Z[0,6]→Z[0,7]→Z[0,8].
- For pair 5:
  DRA and SCARA:
  Z[6,10]→Z[5,10]→Z[4,10]→Z[3,10]→Z[2,10]→Z[1,10]→Z[0,10]→Z[11,13]→Z[10,13]→Z[9,13]→Z[8,13]→Z[8,14]→Z[8,15]→Z[8,16]→Z[8,17]→Z[8,18]→Z[8,19].
  DWCA:
  Z[6,10]→Z[6,11]→Z[7,11]→Z[8,11]→Z[8,12]→Z[8,13]→Z[8,14]→Z[8,15]→Z[8,16]→Z[8,17]→Z[8,18]→Z[8,19].
- For pair 6:
  DRA and SCARA::
  Z[7,6]→Z[7,5]→Z[7,4]→Z[7,3]→Z[6,3]→Z[5,3]→Z[4,3]→Z[3,3]→Z[2,3]→Z[1,3]→Z[0,3]→Z[11,20]→Z[10,20]→Z[9,20]→Z[8,20].
  DWCA:
  Z[7,6]→Z[7,5]→Z[7,4]→Z[7,3]→Z[7,2]→Z[7,1]→Z[7,0]→Z[8,0]→Z[8,23]→Z[8,22]→Z[8,21]→Z[8,20].

From the above result, we can see DWCA has the highest performance to find the shortest path. However SCARA performs better than DRA in most cases.

D. Time Complexity

SCARA and DRA use rules to decide the next hop for each packet. The time need to calculate the direction does not change with the number of satellites. However, Dijkstra algorithm has high time complexity $O(N^2)$ [27], where N stands for the number of satellites. For measuring the precise time used for route a packet, the GetTickCount[28] function is called before and after the routing function. They are referred as $Tick_{before}$ and $Tick_{after}$ respectively. The actually ticks can be calculated by $Tick_{after} - Tick_{before}$. The simulation runs on a DELL OPTIPLEX 755 computer, which has a CPU of Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.4GHz and 2GB RAM.

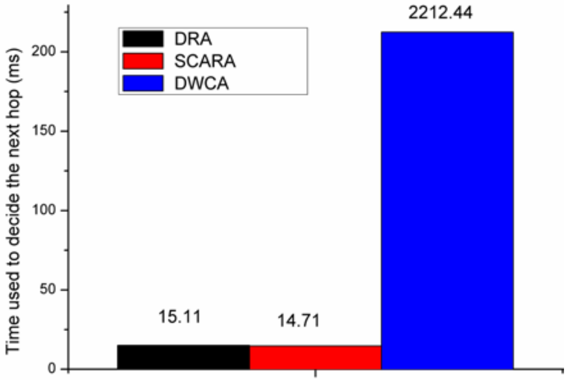

Figure 4. Time Complexity

The simulation result shows that SCARA runs fastest with average ticks of 14.71 for routing deciding a next direction. DRA needs 15.11 average ticks. However, DWCA spends 2212.44 average ticks. For comparing the time used to decide the next hop by DRA, SCARA and DWCA, the number of 212.44, not the real one of 2212.44, is used to draw the bar for DWCA. These numbers prove that the time complexity of SCARA is lowest, while DWCA takes the highest.

VI Conclusion

Routing is a critical task for satellite IP networks. After analyzing the characteristics of LEO constellation carefully, we find some simple rules which can be adopted to produce a new routing algorithm. These rules include: 1) there are two directions applying to reduce the routing path; 2) the direction with higher satellite latitude has the priority; 3) Entering the polar region should be tried to avoid;4) local buffer and queue information is used to avoid congestion, etc. Based on these rules, we presented a new distributed routing algorithm (SCARA) for satellite IP networks. It uses only local information when deciding the next direction and tries to find the shortest routing path. Congestion avoidance capability is also provided by SCARA.

For evaluating the performance, a specialized simulator is developed which implements SCARA, DRA and DWCA routing algorithms. After comparing the results, we find SCARA has better performance than DRA and DWCA. It has the capability to get the minimum path and has lower end-to-end delay.

The source code of our simulator is open and uploaded to the Internet. Everyone can download it from http://www.zuotiwang.com/simulator/scara.zip. Through this way, other researchers can check, test and reuse our simulator. Five months are spent to complete the simulator which includes more than 120 thousand lines of code. We hope this can promote further research in this field.

Acknowledgment

This work is supported in part by Chongqing Engineering Research Center for Instrument and Control Equipment and the Fundamental Research Funds for the Central Universities.

References

[1] Wang, L., L. Liu, and X. Hu, *Load-aware and Position-aided Routing in Satellite IP Networks.* International Journal of Computers Communications & Control, 2012. **7**(1): p. 163-172.
[2] Korçak, Ö. and F. Alagöz, *Virtual topology dynamics and handover mechanisms in Earth-fixed LEO satellite systems.* Computer networks, 2009. **53**(9): p. 1497-1511.
[3] Ganguly, D., et al., *Area based Mobility Management by using GPS in LEO satellite Networks.* International Journal of Computer Applications, 2012. **42**(13): p. 41-43.
[4] Wood, L., et al., *IP routing issues in satellite constellation networks.* International Journal of Satellite Communications, 2001. **19**(1): p. 69-92.
[5] Moy, J., *OSPF version 2.* 1997.
[6] Malkin, G.S., *RIP version 2.* 1998.
[7] Werner, M., et al., *ATM-based routing in LEO/MEO satellite networks with intersatellite links.* Selected Areas in Communications, IEEE Journal on, 1997. **15**(1): p. 69-82.
[8] Chang, H.S., et al., *FSA-based link assignment and routing in low-earth orbit satellite networks.* Vehicular Technology, IEEE Transactions on, 1998. **47**(3): p. 1037-1048.
[9] Uzunalioğlu, H., et al., *Footprint handover rerouting protocol for low earth orbit satellite networks.* Wireless Networks, 1999. **5**(5): p. 327-337.
[10] Uzunalioglu, H. *Probabilistic routing protocol for low earth orbit satellite networks*. 1998: IEEE.
[11] Yin, Z., L. Zhang, and X. Zhou, *On-Demand QoS Multicast Routing for Triple-Layered LEO/HEO/GEO Satellite IP Networks.* Journal of Communications, 2011. **6**(6): p. 495-508.
[12] Chang, H.S., et al. *Performance comparison of static routing and dynamic routing in low-earth orbit satellite networks*. 1996: IEEE.
[13] Lu, Y., et al., *Dynamic Fault-tolerant Routing based on FSA for LEO satellite networks.* IEEE Transactions on Computers, 2012.
[14] Akyildiz, I.F., E. Ekici, and M.D. Bender, *MLSR: a novel routing algorithm for multilayered satellite IP networks.* Networking, IEEE/ACM Transactions on, 2002. **10**(3): p. 411-424.
[15] Chen, C., E. Ekici, and I.F. Akyildiz. *Satellite grouping and routing protocol for LEO/MEO satellite IP networks*. 2002: ACM.
[16] Taleb, T., et al. *Tailoring ELB for multi-layered satellite networks*. 2009: IEEE.
[17] Henderson, T.R. and R.H. Katz. *On distributed, geographic-based packet routing for LEO satellite networks*. 2000: IEEE.
[18] Ekici, E., I.F. Akyildiz, and M.D. Bender, *A distributed routing algorithm for datagram traffic in LEO satellite networks.* Networking, IEEE/ACM Transactions on, 2001. **9**(2): p. 137-147.
[19] De Sanctis, M., E. Cianca, and M. Ruggieri. *IP-based routing algorithms for LEO satellite networks in near-polar orbits*. 2003: IEEE.

[20] Gounder, V.V., R. Prakash, and H. Abu-Amara. *Routing in LEO-based satellite networks*. 1999: IEEE.
[21] Alagoz, F., O. Korcak, and A. Jamalipour, *Exploring the routing strategies in next-generation satellite networks.* Wireless Communications, IEEE, 2007. **14**(3): p. 79-88.
[22] Yan, J., Y. Zhang, and Z. Cao, *Reverse Detection Based QoS Routing Algorithm for LEO Satellite Constellation Networks*.* Tsinghua Science & Technology, 2011. **16**(4): p. 358-363.
[23] Jiang, W. and P. Zong, *A Discrete-Time Traffic and Topology Adaptive Routing Algorithm for LEO Satellite Networks.* International Journal of Communications, Network and System Sciences, 2011. **4**(1): p. 42-52.
[24] Chen, C. and E. Ekici, *A routing protocol for hierarchical LEO/MEO satellite IP networks.* Wireless Networks, 2005. **11**(4): p. 507-521.
[25] Xiren, X., *Computer networks*. 4 ed. 2006: Publishing House of Electronics Industry.
[26] Qiu, T., et al., *A Failure Self-recovery Strategy with Balanced Energy Consumption for Wireless Ad Hoc Networks.* Journal of Computers, 2012. **7**(1): p. 116-123.
[27] Wang, M., *The New Algorithm for Finding the Paths based on Coding Graph.* Journal of Software, 2012. **7**(1): p. 1-8.
[28] Zhanqi, H. and D. Feiyue. *Study on method of precision timing for open CNC systm based on windows*. 2010: IEEE.

Liu Feng born in 1957, he is currently a professor in School of Computer and Information Science, Southwest University, China. His main research interests include automation systems, communication networks, fieldbus device integration, etc.

Zhang Yu born in 1979, he is currently an associate professor in School of Computer and Information Science, Southwest University, China. His main research interests include communication networks, fieldbus device integration, space monitoring and communication, etc.