ETER: Encounter Time Estimating Routing for Satellite IP Networks

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

***Abstract*—Satellites in Satellite IP network (SIPN) run along its orbit and have fixed periods. It is possible to estimate the encounter time between two satellites using their history encountering information. We introduce a method to estimate the time. Based on the encounter time information estimated, a new routing method, named Encounter Time Estimating Routing (ETER), is presented in the paper. ETER firstly estimates the encounter time among satellites. Secondly, it forms a graph, assign values of encounter time as the weight of edges. Then ETER path algorithm, a modified Dijkstra algorithm, finds the best routing set for packets. Finally, ETER forward each packet using ETER forwarding algorithm. The simulation result shows that ETER has better performance on packet loss rate and transferring throughput under all traffic loads; under lower traffic loads, the average end-to-end delays of ETER is a little higher than Epidemic routing, but lower than PKR; under higher traffic loads, the delays is lower than PKR and close to Epidemic routing.**

***Index Terms*—Encounter Time Estimating Routing; Satellite IP Networks; Epidemic Routing; DTN**

I. Introduction

Satellite IP network (SIPN) is a network using satellite to transfer Internet packets. With the great popularity of the Internet and the rapid development of the next generation network in terrestrial networks, satellite networks will be required to provide connectionless service and transport IP-based traffic[1]. SIPN is a logical extension of terrestrial based internetworks and rapidly gaining interest in both commercial and military applications[2]. Fig.1 shows a satellite IP network. Each satellite moves along its orbit. The orbits can be classified according to the orbits' altitude as 1) The Low Earth orbit (LEO) with altitudes up to 2,000 km; 2) The Medium Earth orbit (MEO) with orbits ranges in altitude from 2,000 km to just below geosynchronous orbit at 35,786 km; 3) The High Earth orbit with orbits above the altitude of geosynchronous orbit 35,786 km. Two satellites may have chance to talk when they get close enough, i.e. locate in a communication intersection.

Satellite IP network has the characteristics of 1) a satellite moves like randomly to others; 2) the transmitting delay from one satellite to another is very high; 3) each satellite not only capture and disseminate its own IP packets, but also serve as a relay for other satellites. The network is running like a mesh network; 4) it is hard for a satellite to set up an efficient path when routing IP packets.

From the point of a satellite, without the orbit information of another's, the encounter time with another one in a communication intersection is unknown. Unfortunately, in SIPN, a satellite always knows nothing about other satellites' orbit. Because 1) a satellite in one constellation may know nothing about another one in other constellations; 2) a satellite has no knowledge about another one which launched later; 3) a satellite loses the update information about another one whose orbit has been changed; 4) a satellite may have limited memory, which makes it impossible to save all orbits information of others; 5) the orbits information may be confidential, and are only open to some satellites. So other satellites can not get such information. Without the orbit information, one satellite's moving pattern seems randomly to others. The routing methods for terrestrial network are no longer effective in SIPN.

SIPN is a kind of Delay- and Disruption-Tolerant Networking (DTN)[3], which provides communication in highly stressed environments such as variable delays, discontinuous connectivity and high bit error rate[4]. DTNs have the characteristic such as the mobilization model is heterogeneous; the end-to-end connectivity may not exist; the computing capabilities are different; the buffers and other resources are different. Traditional routing method for terrestrial Internet does not meet the requirement of DTN. Open Shortest Path First (OSPF)[5], Intermediate System to Intermediate System (IS-IS)[6] and Optimized Link State Routing (OLSR)[7] are used in traditional Internet. They flood topology changes to all nodes in a routing area to maintain link states and to avoid routing loops. But in DTNs, maintaining link states is something practically impossible. There has no known end-to-end route in DTNs. So Distance Vector protocols such as Border Gateway Protocol (BGP) and Enhanced Interior Routing Gateway Protocol (EIRGP), which have built-in routing loop detection mechanisms, can not be applied in SIPN either[8].

Traditional routing methods in [1, 9-15] for SIPN treat the network architecture as one or several layers and handle differently. The satellites in different layer have different roles and provide service for different application. They are not regarded as equivalent nodes with same communication role. Furthermore intraplane and interplane links are always assumed to exist all the

doi:10.4304/jsw.9.2.474-481

time. However, the network is not considered to be a DTN network. The concepts and methods in DTN, like in [16, 17], are not combined with these routing methods in satellite IP network.

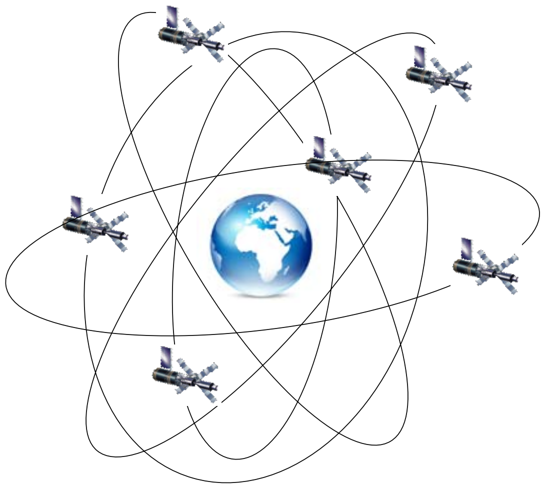

Figure 1. Satellite IP network

We found that the packet loss rate may be high and the throughput may be low, when Epidemic is adopted as the routing method for Satellite IP network. Since the buffers in satellite are very limited. Epidemic may produce more copies of packet. At the time of all the buffers are used, the coming of new packets will overwrite the packet waiting in queue. This may lead to the loss of the packet. We also found that the time of two satellites entering a communication intersection may be estimated using history information. Based on the information, we introduce a novel estimating algorithm and a new Encounter Time Estimating Routing (ETER) method for SIPN routing.

The paper is organized as follows. Section II presents the related research work in this field. Section III gives introduction to satellite orbit. The encounter time estimating algorithm is illustrated in Section IV. Based on the estimating algorithm, Section V introduces a new routing method Encounter Time Estimating Routing. The performance evaluation and analysis routing is presented in Section VI. Finally, Section VII concludes the paper.

II. Related Work

Routing strategies for IP or IP-like switches on-board satellites have already been extensively studied[1]. Some routing methods for one layer satellite IP network are introduced in [12-15]. Geographic-based addresses are used and a distributed routing protocol is proposed in [12] for Low-Earth-Orbiting (LEO) satellite constellations. The next hop is selected based on a minimization of the remaining distance to the destination. The datagram routing algorithm (DRA)[13] introduces the concept of logical locations of the LEO satellites and presents a routing method to forward the packets with the minimum propagation delay.

Some routing methods for multi-layered satellite IP network are also studied as in [1, 9-11]. Multi-Layered Satellite Routing (MLSR) [9] and Satellite Grouping and Routing Protocol (SGRP) [10] are presented for multi-layered satellite IP network. MLSR algorithm is introduced for a satellite IP network consisting of LEO, medium-Earth orbit (MEO), and geostationary Earth orbit (GEO) satellites. MLSR calculates routing tables efficiently using the collected delay measurements periodically. SGRP is developed in [10]. In each snapshot period, SGRP divides Low Earth Orbit (LEO) satellites into groups according to the footprint area of the Medium Earth Orbit (MEO) satellites. Based on the delay reports sent by LEO satellites, MEO satellite managers compute the minimum-delay paths for their LEO members.

Unicasting routing methods for satellite IP network are explored in [9, 10, 12, 13]. However, with the growth of Internet-based multimedia applications, multicasting constitutes an important service to perform the simultaneous distribution of the same multicasting packets from a single source node to a group of destinations in the satellite IP networks[1]. A multicast routing algorithm for LEO satellite IP networks is introduced in [14]. It uses the DRA to create the multicast trees, which minimizes the end-to-end delay for real time multimedia services. For the wireless bandwidth in satellite networks is a limited and scarce resource, [15] presents a bandwidth-efficient multicast routing mechanism to minimize the total bandwidth(the number of hops). It formulates the problem as an optimization problem and proposes a distributed algorithm and a protocol to support the dynamic group membership for multicasting over LEO satellite networks. A distributed multicast routing scheme is introduced in [11] for multi-layered satellite IP networks. It aims to minimize the total cost of multicast trees in the satellite network. Multicast trees are constructed and maintained in the dynamic satellite network topology in a distributed manner. On-demand QoS multicast routing protocol (ODQMRP) for a triple-layered LEO/HEO/GEO satellite network architecture is presented in [1]. In ODQMRP, the link state information piggybacked by each satellite is exchanged through the link state report process, and the network topology is acquired by the source node through the route discovery and route reply process. The PSPT and LCT strategies are used to minimize the path delay and the path cost of the multicast trees, respectively.

Many researches carry out research on DTNs routing. In [16], the developments in this area is addressed into three different sub-categories: Message Ferrying Approach, Inter-Region Routing and Multicast Routing. Multicast routing methods, like Epidemic routing [17], show good performance in DTN. Epidemic routing relies upon the transitive distribution of messages through ad hoc networks. Each host maintains a buffer consisting of messages that it has originated as well as messages that it is buffering on behalf of other hosts. When two hosts come into communication range of one another, the host with the smaller identifier initiates an anti-entropy session.

During anti-entropy, the two hosts exchange their summary vectors to determine which messages stored remotely have not been seen by the local host. In turn, each host then requests copies of messages that it has not yet seen.

In this paper, we take routing methods in traditional satellite network and DTN into account at the same time.

III. SATELLITE ORBIT

The motion pattern of a satellite is decided by its orbit. Fig. 2 illustrates the Keplerian orbital elements and a satellite position[18]. The satellite's position in an ideal, non-perturbed orbit can be represented by:

- size and shape of the ellipse : semi-major axis a and eccentricity e;
- orientation of the orbital plane relative to the Earth : orbit inclination I and longitude of the ascending node Ω;
- orientation of the ellipse in the orbital plane : argument of perigee ω;
- satellite position in the ellipse : true anomaly U; and
- a reference time when passes the perigee : t_p.

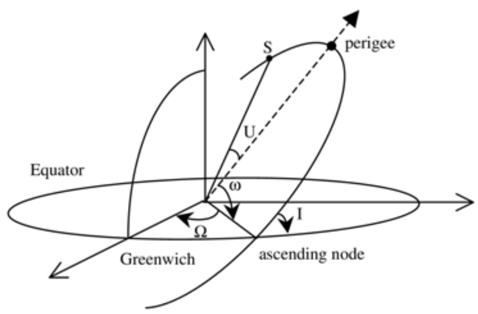

Figure 2. Keplerian orbital

The "natural" satellite orbital plane system is defined by the origin located at one focus of the elliptical orbit, which corresponds to the position of the mass centre of the Earth. The x-axis and y-axis are coincident with the major and minor axes of the orbital ellipse respectively. Fig. 3 illustrates the natural orbital plane system where the positive x-axis passes through perigee. A satellite position in the natural orbital plane system can be expressed by Eq.(1), where a is the semi-major axis of the satellite orbit ; e is the eccentricity of the orbit ; E is the orbital eccentric anomaly ; r is the instantaneous distance between the satellite and the centre of the Earth ; and U is the true anomaly.

$$\vec{r} = \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} = \begin{pmatrix} a\cos E - a\cdot e \\ a\sqrt{1-e^2}\sin E \\ 0 \end{pmatrix} = \begin{pmatrix} r\cos U \\ r\sin U \\ 0 \end{pmatrix} \quad (1)$$

Using the coordinates x and y, which denote the satellite's position in the orbital plane with respect the center of Earth, one may express the real velocity $h = |\mathbf{h}|$ as a function of

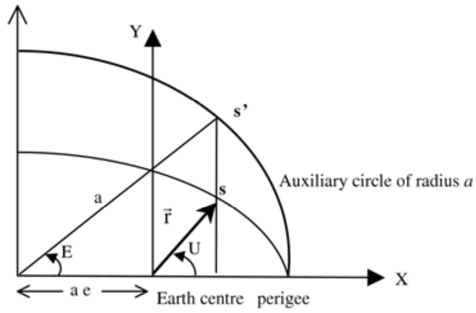

Figure 3. The natural orbital plane system

E : $h = x\cdot\dot{y} - y\cdot\dot{x} = a^2\sqrt{1-e^2}\dot{E}(1-e\cos(E))$ [19] .

This equation may further be simplified using $h = \sqrt{GM_\oplus a(1-e^2)}$ to give the following differential equation for the eccentric anomaly:

$$(1-e\cos E)\dot{E} = n .$$

Here the mean motion $n = \sqrt{GM_\oplus / a^3}$ has been introduced to simplify the notation. $GM_\oplus$ is the gravitational coefficient, i.e. the product of the gravitational constant and the Earth's mass. It has been determined with considerable precision from the analysis of laser distance measurements of artificial Earth satellites:

$$GM_\oplus = 398600.4405 \pm 0.001 km^3 s^{-2} .$$

Integrating with respect to time finally yields Kepler's equation $E(t) - e\sin E(t) = n(t-t_p)$, where t_p denotes the time of perigee passage at which the eccentric anomaly vanishes. The right side $M = n(t-t_p)$ is called the mean anomaly. It changes by 360° during one revolution but , in contrast to the true and eccentric anomalies, increases uniformly with time[19]. The orbital period is proportional to the inverse of the mean motion n and is given by

$$T = 2\pi / n = 2\pi\sqrt{a^3 / GM_\oplus} \quad .$$

From Fig. 2, it can be seen that the transformation of the Cartesian components of a satellite position vector $\vec{r}$ from the orbital plane coordinate system to the ECEF system (Earth-Centered, Earth-Fixed coordinate system[20]) may be carried out by three rotations in the following order:

- a first rotation by the argument of the perigee ω;
- a second rotation by the angle of inclination I; and
- finally a rotation by the angle of the longitude of ascending node Ω.

The corresponding transform equation is calculated by Eq.(2), where $R_n(\theta)$ is the rotation matrix, the subscript $n = 1,3$ corresponding to the rotation axes of x, z respectively.

$$\vec{r}_{ECEF} = R_3(-\Omega)\cdot R_1(-I)\cdot R_3(-\omega)\vec{r} \quad (2)$$

The rotation matrixes $R_3(\theta)$ is expressed in the forms of

$$R_3(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The $R_1(\theta)$ is

$$R_1(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix}$$

IV. Encounter Time Estimation Method

Satellites in SIPN move along their orbits. Only when the distance between two satellites is short enough, one can receive the signal from another one. Then they have the chance to exchange packets. Most satellites have elliptical orbits. Considering two satellites' orbits in the space, there may be zero, one, two, or more chances to talk directly between the two satellites. As shown in Fig. 4 a), there is no chance exists for the two satellites running along the orbits, i.e. the two satellites can not communicate directly. Fig.4 b) shows that one communication intersection exists for the two satellites along their orbits. When both satellite entered the intersection, they may exchange packets directly. Fig.4 c) and d) show their may have more than one communication intersection for two satellites.

If all parameters of satellites S_a orbit are known by satellite S_b, S_b may get the time of next encounter with S_a in a communication intersection by calculation. For S_b knows orbit parameters of S_a, it can compute S_a's position using Equation(2). At the same time, S_b has its own orbit parameters and its current position information. With this information S_b can compute the time when they will meet in a communication intersection. On the other side, if S_a knows the orbit parameters of S_b, it can also get the encountering time by calculation.

Furthermore, with orbit parameters, a satellite can also compute how many communication intersections exist and when two satellites will meet in the intersections. So in this case, when routing packets, a satellite may find a best way to route packets efficiently.

However, as analyzed in Section I, in satellite networks, a satellite always has no information about other satellites' orbit parameters. There is no possibility to use orbit parameters to compute the encountering time directly. Satellite periodically runs along its orbit. So it is possible for a satellite to estimate the encounter time with others using history information. In this paper, historical encounter time t_i is used to predict the next encounter time in $Region_i$. Variable t_i denotes the time spend from last departure to encounter again in the same communication intersection $Region_i$.

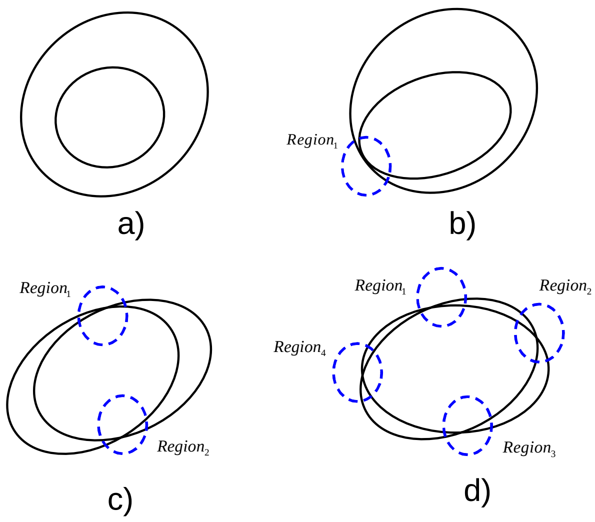

Figure 4. The Communication Intersections between two Satellites

As shown in Fig.4, there may exist zero, one, two or more communication intersections between two satellites. For the case of one communication intersection, the estimation equation (3) is used.

$$t_{estimated} = t_1 \qquad (3)$$

For the case of more than one communication intersections, the estimation equation (4) is used.

$$t_{estimated} = c_1t_1 + c_2t_2 + ... + c_nt_n = \sum_{i=1}^{n} c_it_i \qquad (4)$$

where n, $n = 1, 2, ...$, is the number of communication intersections exist between two satellites, c_i, $1 \le i \le n$, $\sum c_i = 1$, is the weighting coefficient. For different SIPN, c_i should be different and should be adjusted to get expected performance.

V. Encounter Time Estimating Routing

A. Calculating Encounter Time

Equation (4) is used to estimate the encounter time between two satellites. Algorithm 1 shows the calculating procedure. The total number of communication intersections is set to $n = 4$. The initial weighting coefficients are set to $c_i = 0.25$ and the time parameters are set to $t_i = \infty$, where $i = 1, 2, 3, 4$. If satellite S_i and S_j entered a communication region i, the weighting coefficients are updated according to

$$\begin{cases} c_j = c_j * 0.9, & j \ne i \\ c_j = 1 - \sum_{j \ne i} c_j, & j = i \end{cases}$$

Algorithm 1: Calculating encounter time

//Setp1: Initialize all parameters
$c_i = 0.25, t_i = \infty \quad i = 1, 2, 3, 4$
//Step2: calculate weights and time
for (all satellite pairs $< i, j >$)
    if (satellite i and j just entered a comm. region)
        if (meet in the first comm. region)
            $c_j = c_j * 0.9, \quad j \ne 1$;

$c_1 = 1 - \sum_{j \neq 1} c_j$;
Calculate t_1 ;
else if (meet in the second comm. region)
$c_j = c_j * 0.9$, $j \neq 2$;
$c_2 = 1 - \sum_{j \neq 2} c_j$;
Calculate t_2 ;
else if (meet in the third comm. region)
$c_j = c_j * 0.9$, $j \neq 3$;
$c_3 = 1 - \sum_{j \neq 3} c_j$;
Calculate t_3 ;
else
$c_j = c_j * 0.9$, $j \neq 4$;
$c_4 = 1 - \sum_{j \neq 4} c_j$;
Calculate t_4 ;
endif
endif
endfor
//Step 3: Estimate the encounter time
for (all satellite pairs $< i, j >$)
$t_{estimated} = c_1 t_1 + c_2 t_2 + ... + c_4 t_4 = \sum_{i=4}^{n} c_i t_i$
endfor

B. Mapping to a Graph

The satellites are mapped to nodes of a graph. After the estimation of all $t_{estimated}[i, j]$, which denotes the estimated encounter time for satellite i and j , are finished, $t_{estimated}[i, j]$ is assigned to $E[i, j]$. $E[i, j]$ is the weight of edge from node i to j . Finally, a bidirectional graph with known edge weigh is formed, as shown in Fig. 5.

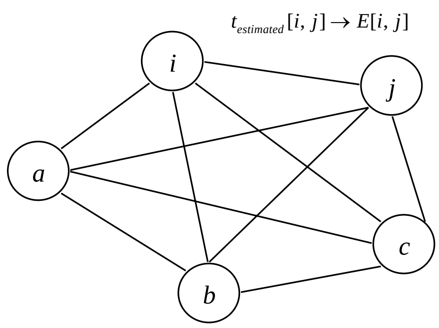

Figure 5. Graph mapping

The procedure of mapping is shown in Algorithm 2.

Algorithm 2: Mapping to a graph

Variable $t_{infinite} = \infty$;
for (all satellite pairs $< i, j >$)
if ($i \neq j$)
$E[i, j]$:= Min($t_{estimated}[i, j]$, $t_{infinite}$);
else
$E[i, j]$:= $t_{infinite}$;
endif;
endfor

C. Finding Node Sets

ETER algorithm is based on Dijkstra shortest path algorithm with some modifications. There are two main differences from Dijkstra algorithm.

1) ETER algorithm outputs a set of nodes

Dijkstra algorithm will output a shortest path between any two nodes. For satellite S_j and S_j , the path may looks like $S_i \rightarrow S_2 \rightarrow S_3 \rightarrow S_j$. While ETER algorithm outputs a set like $SET_{ij} = [2, 3, j]$.

2) ETER algorithm outputs a null set, i.e. $SET_{ij} = [\ ]$, if the total weight of the path $\sum E_{ij} = \infty$.

For example, if Dijkstra algorithm outputs a path $S_i \rightarrow S_2 \rightarrow S_3 \rightarrow S_j$, and the total weight $E_{i2} + E_{23} + E_{3j} = \infty$, then ETER will output a null set.

The algorithm is illustrated in Algorithm 3.

D. Forwarding Packets

Each Satellite will keep on checking whether another one is in the communication region. When two satellites meet in the communication region, they use following procedure to route packets.

Satellite S_i finds the oldest packet P_i which did not send to Satellite S_j before;

Satellite S_j finds the oldest packet P_j which did not send to Satellite S_i before;

As soon as the link from S_i to S_j is available, S_i sends P_i to S_j and marks P_i has been sent to S_j ; when Satellite S_j received P_i , it checks whether it has been save in the buffer. If yes, desert P_i ; if no, save P_i into the buffer;

When the link from S_j to S_i is available, Satellite S_j sends P_j to S_i and marks P_j has been sent to S_i ; When Satellite S_i received P_j , it checks whether it has been save in the buffer. If yes, desert P_j ; if no, save P_j into the buffer.

Algorithm 2 ETER algorithm

```
//  SolvedSet, UnSolvedSet: the sets of points which have been handled and not handled  by ETER algorithm
// PkrSet: the set of points outputted  by ETER algorithm
// EdgeWeight: the weight of edge in the graph
// DynamicWeight: the weight from the source point to current point
// PointsNum: the number of points in the graph
// StartPoint: the point specified by user
//initialize
Put StartPoint into SolvedSet;
Delete StartPoint from UnSolvedSet;
Put StartPoint into PkrSet;
Set PkrPathWeight[StartPoint] to 0;
NextPoint := 0;
//while
while UnSolvedSet is not null do
   //find a point(NextPoint) in UnSolvedSet which has the smallest weight
   MinWeight := +∞;  // MinWeight: a temperate real variable
   for i := 0 to PointsNum - 1 do
      if i is in UnSolvedSet then
         if DynamicWeight[i] <= MinWeight then
           MinWeight := DynamicWeight[i];
           NextPoint := i;
         endif;
      endif;
   endfor;
  // Update the weight
   for i := 0 to PointsNum - 1 do
      if i is in UnSolvedSet then
         if (DynamicWeight[NextPoint] + EdgeWeight[NextPoint, i]) < DynamicWeight[i] then
           DynamicWeight[i] := DynamicWeight[NextPoint] + EdgeWeight[NextPoint, i];
         endif;
         PkrPathWeight[i] := DynamicWeight[i];
      endif;
   endfor;
   // find the set
   for i := 0 to PointsNum - 1 do
      if i  is in SolvedSet then
        if Round(PkrPathWeight[i] + EdgeWeight[NextPoint, i]) = Round(PkrPathWeight[NextPoint]) then
             PkrSet[NextPoint] := PkrSet[i] + [NextPoint];
        endif;
      endif;
   endfor;
   //update SolvedSet and UnSolvedSet
   Put NextPoint into SolvedSet;
   Delete NextPoint from UnSolvedSet;
 endwhile;
```

VI. SIMULATION

ETER simulator is developed for performance evaluation. ETER routing is implemented in the simulator. Epidemic and PKR[21] routing methods are also included for performance comparation. There are 6 satellites in the satellite IP network simulated. The parameters of satellites' orbit information are shown in Table 1.

First in first out (FIFO) queue is used in the simulation. When the buffer is fully used, the coming of a new packet will overwrite the oldest packet in the buffer. If all copies of the overwritten packet have not reach the destination, the packet will be market as a loss.

TABLE 1
SATELLITE ORBIT PARAMETER

Satellite orbit parameters	**1**	**2**	**3**	**4**	**5**	**6**
semi-major axis a (km)	12000	13000	14000	15000	16000	17000
eccentricity e	0.6	0.1	0.09	0.6	0.31	0.2
orbit inclination I (°)	0.1667	0.0278	0.1977	0.1861	0.1194	0.2222
longitude of the ascending node Ω (°)	0.0833	0.0833	0.0833	0.0833	0.0833	0.1056
argument of perigee ω (°)	0.1139	0.1722	0.2222	0.1139	0.0306	0.2083
reference time t_p	36	2	10	36	3	26

The simulation is carried out under different traffic load $n, n = 1, 2, ..., 12$. Where n is the number of packet(s) from one to others satellite in each 10 minutes. For example, for traffic load 5, there are 5 packets generated from S_i to S_j, $j \neq i$, $j \in S_{all}$ every 10 minutes, where S_{all} is the universal set of satellites.

The simulation time is set to 48 hours or two days.

A. Simulation result - Loss Rate

The packet loss rate of ETER is lower than Epidemic and PKR routing methods, as shown in Fig. 6. A packet will be regarded as loss when it does not reach to the destination and all the copies have been deleted in satellites' buffer.

ETER estimates the next encounter time of two satellites. The time information is useful to search the efficient routing path. A modified Dijkstra algorithm is used to calculate such a routing path. All the nodes along the path constitute a routing set. ETER finally output the set as the result.

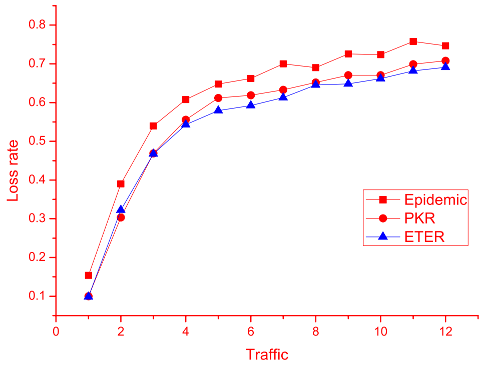

Figure 6. Loss rate

When forwarding the packets, ETER will check whether the satellite set is null. If yes, ETER forwards the packets to all satellites met. If not, ETER only forwards the packets to the satellites in satellite set. By this mean, fewer packet copies are generated and reduce the possibility to overwrite other packets. So packets can get more time and chances to reach its destination. It can be seen easily in Fig. 7, ETER has lower loss rate than Epidemic routing under each traffic load. Comparing to Epidemic routing, it can be drawn that ETER should be used if lower loss rate is required.

B. Simulation Result - Throughput

From Fig.7, it is known that ETER has higher throughput than PKR and Epidemic routing. Under each traffic load simulated, the number of reached packets of ETER is bigger than Epidemic and PKR routing methods.

PKR generate fewer packet copies than Epidemic routing. The chance for each packet to reach its destination is higher. PKR uses orbit information partially known to decide the routing path. For these satellites with known orbit information, PKR can route their packets efficiently with fewer copies and lower loss. For satellites without known orbit information, PKR works as normally flooding routing does.

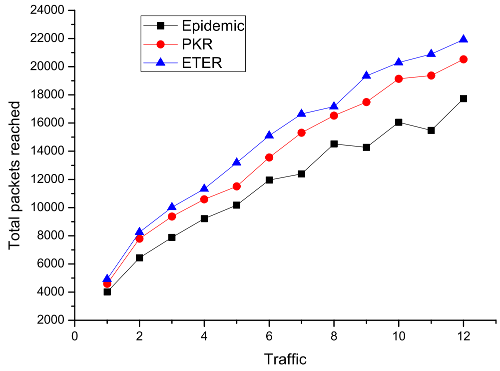

Figure 7. Throughput

ETER runs like PKR. For it can estimate the next encounter time between satellites in an efficient way, the best routing set may be found using the time information. The packets routed by ETER have same possibility to reach their destination as PKR and Epidemic do. However ETER generates fewer packet copies, the overview queue time in the buffer is lower than PKR and Epidemic. Therefore ETER can route more packets to their destination.

C. Simulation Result - Average Delay

As shown in Fig.8, under lower traffic load, the average end-to-end delay of ETER is a little higher than Epidemic routing, but lower than PKR. Under higher traffic load, the average end-to-end delay performance of ETER is close to Epidemic routing.

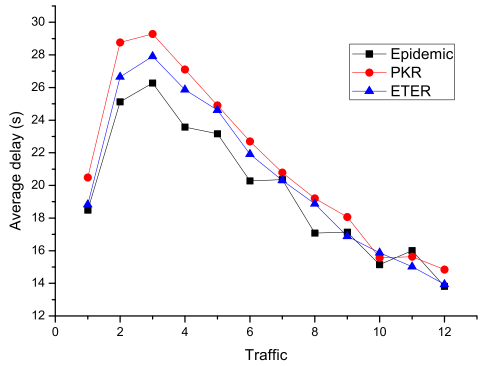

Figure 8. Average delay

Epidemic routing forwards all packets to the satellites it met. So the number of packet copies is increasing scientifically with the number of nodes in the network.

Under lower traffic load, more copies of a packet help to reach its destination timely. ETER and PKR get higher average end-to-end delay than Epidemic routing. ETER shows higher performance than PKR.

Under higher traffic load, ETER and PKR perform similarly as Epidemic routing, as the shown in the right part of the curve in Fig. 8 The buffer of the satellites is fully filled because the high traffic loads lead to generate more packets in the queues. Packets forwarded by ETER, PKR or Epidemic routing are suffered high average end-to-end delays. With the traffic loads going higher, the better PKR performs when comparing with Epidemic routing.

VII. CONCLUSION

Routing method is a key part of satellite IP networks of routing. It is found in this paper that, as a kind of Delay Torrent Network, satellite's motion pattern may be estimated using its orbit information.

PKR uses orbit information of part satellites to estimate next meeting time between satellites. Then the meeting time is mapped to the edge weight of a graph. The satellite set which includes all satellites in the shortest path is outputted by ETER algorithm. Finally, ETER forwards packets based on the satellite set. If the satellite set is not null, ETER will forward one copy of a packet to only one satellite in the satellite. Otherwise, packets will be flooded to other satellites.

From the simulation result, it can be seen that ETER has better performance than PKR and Epidemic on packet loss rate and transferring throughput under all traffic loads; the average end-to-end delays of ETER is lower than PKR and a little higher than Epidemic routing under lower traffic loads, but the delays is close to and even better than Epidemic routing under higher traffic loads. It can be drawn that ETER can be used when higher loss rate and throughput are required, or when routing with high traffic load.

The source code of our simulator is open and uploaded to the Internet. It can be downloaded at http://www.zuotiwang.com/simulator/eter.zip. Through this way, other researchers may check our simulator, reproduce the simulating, and reuse the source code for 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.

REFERENCE

[1] 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.
[2] Johnson, J.D., et al. *Internet routing in space NMS architecture*. 2009: IEEE.
[3] Fall, K. and S. Farrell, *DTN: an architectural retrospective.* Selected Areas in Communications, IEEE Journal on, 2008. 26(5): p. 828-836.
[4] Yanggratoke, R., et al., *Delay Tolerant Network on Android Phones: Implementation Issues and Performance Measurements.* Journal of Communications, 2011. 6(6): p. 477-484.
[5] Moy, J., *OSPF version 2.* 1997.
[6] Smit, H. and T. Li, *Intermediate system to intermediate system (IS-IS) extensions for traffic engineering (TE).* 2004.
[7] Jacquet, P., *Optimized link state routing protocol (OLSR).* 2003.
[8] Khabbaz, M., C. Assi, and W. Fawaz, *Disruption-Tolerant Networking: A Comprehensive Survey on Recent Developments and Persisting Challenges.* Communications Surveys & Tutorials, IEEE, 2011(99): p. 1-34.
[9] 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.
[10] Chen, C. and E. Ekici, *A routing protocol for hierarchical LEO/MEO satellite IP networks.* Wireless Networks, 2005. 11(4): p. 507-521.
[11] Akyildiz, I.F., E. Ekici, and G. Yue, *A distributed multicast routing scheme for multi-layered satellite IP networks.* Wireless Networks, 2003. 9(5): p. 535-544.
[12] Henderson, T.R. and R.H. Katz. *On distributed, geographic-based packet routing for LEO satellite networks*. 2000: IEEE.
[13] 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.
[14] Ekici, E., I.F. Akyildiz, and M.D. Bender, *A multicast routing algorithm for LEO satellite IP networks.* Networking, IEEE/ACM Transactions on, 2002. 10(2): p. 183-192.
[15] Yang, D.-N. and W. Liao, *On multicast routing using rectilinear Steiner trees for LEO satellite networks.* Vehicular Technology, IEEE Transactions on, 2008. 57(4): p. 2560-2569.
[16] Zhang, Z. and Q. Zhang, *Delay/disruption tolerant mobile ad hoc networks: latest developments.* Wireless Communications and Mobile Computing, 2007. 7(10): p. 1219-1232.
[17] Vahdat, A. and D. Becker, *Epidemic routing for partially connected ad hoc networks*. 2000.
[18] Zhang, J., et al., *GPS satellite velocity and acceleration determination using the broadcast ephemeris.* Journal of Navigation, 2006. 59(2): p. 293-306.
[19] Montenbruck, O. and E. Gill, *Satellite orbits: models, methods and applications*. 2005: Springer.
[20] Leick, A., *GPS satellite surveying*. 2003: Wiley.
[21] Zhang, Y. and F. Liu, *Partially Known Routing for Satellite IP Networks* Journal of Software, 2013. 8.

Yu Zhang 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.

Feng Liu 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.