Agent-Based Load Balancing and QoS Routing for Leo Satellite Networks
Ning Li, Lang Hu, Zhong-liang Deng and Tong Su

ABSTRACT: This paper proposes and evaluates the Agent-based load balancing and QoS routing (ALBQR) in the low earth orbit (LEO) satellite networks. Two kinds of agents are designed. Stationary agents located at the satellites calculate link cost using the sum of propagation and queuing delays, employ FARIMA model to predict the traffic of satellite node, modify the link cost based on the predicted traffic. Mobile agents migrate among satellites to gather link status information, notify stationary agents to update routing tables and reserve resource. Mobile agents carry user’s delay, bandwidth and packet loss rate, explore the routing path of satisfy the lowest path cost and QoS constraints. Through simulations on an Iridium satellite network, ALBQR is shown to achieve a balanced traffic distribution among satellites, provide better throughput, decrease packet loss and average end-to-end delay.

INTRODUCTION

Due to the possibility of offering a solution for broadband access to a large geographical coverage, satellite networks are expected to be an essential component of the next-generation network [1]. Differ from terrestrial networks, LEO satellite network has the characteristics of high dynamic, frequent handover, imbalanced traffic load and the interrupt of inter-satellite links [2]. The RIP (Routing Information Protocol) and OSPF (Open Shortest Path First) protocols in terrestrial networks cannot be applied to LEO satellite networks [3]. Therefore, the routing problem is a key issue that needs to be solved urgently in the LEO satellite network.

The goal of load-balancing routing is to let the path goes through low traffic load satellite node. The current load-balancing routing scheme includes [4]: Datagram Routing Algorithm (DRA) and Explicit Load Balancing (ELB). The DRA makes full use of the physical characteristics of the polar orbit constellation itself and does not require on-board processing overhead and signaling overhead, but it is not robust enough to handle satellite node failures and link failures. The ELB completely relies on the busy state advertisement. When the network in high traffic load, many busy state advertisements will further aggravate the network burden.

The goal of QoS routing is to consider delay, bandwidth and packet loss rate requirements [5]. Existing satellites network QoS routing scheme includes: Probabilistic ISL Routing (PIR) [6] and Multi-path QoS routing (MPQR) [7]. PIR only uses the inter-satellite link propagation delay as the path metric, ignoring the
dynamic change of network traffic. The MPQR calculates the optimal path that satisfies both delay and bandwidth constraints. Due to MPQR is based on a genetic algorithm, higher requirements are imposed on the satellite computational power.

The remainder of this paper is organized as follows: In Section 2 describes satellite network architecture and the goal of load balancing and QoS. In Section 3, the new routing scheme called ALBQR is described in detail. Section 4 presented the performance evaluation of ALBQR. Finally, Section 5 concludes this paper.

SYSTEM MODEL

Satellite Network Architecture

We consider an Iridium satellite constellation network. The Iridium satellite constellation network has a total of 66 satellites, 6 polar orbits. Each polar orbit has 11 satellites. The orbit altitude is 799km. Each satellite has four ISLs: two intraplane ISLs and two interplane ISLs. The intraplane ISLs are always maintained. However, due to the periodic motion of the satellite, the interplane ISLs will disconnect in high latitude region (named polar region boundary, typically set to $70^\circ$) [7].

Each satellite has two kinds of agents: stationary agents and mobile agents. Stationary agents are responsible for updating probability routing items and reserving resource. Mobile agents include forward mobile agents and backward mobile agents. When the forward mobile agents migrating to destination satellite, forward mobile agents transform to backward agents and migrating to source satellite with opposite direction. Besides, there exist two kinds of mobile agents: one called normal agent responsible for gathering routing information and notifying stationary agents to update routing table, another named QoS mobile agents take charge of exploring candidate QoS routing paths. The satellite agent platform can be seen in Figure 1.

![Agent Platform](image)

Figure 1. Satellite agent platform.
The goal of Load Balancing and QoS

The goal of load balancing routing is steering traffic to lighter load satellite. We define: For satellite node s goes through the path: $s \rightarrow v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_m \rightarrow v_d$ to the node d. The path cost $PathCost_{s,d}^{v_1}$ of path $(s,d)$ can be defined as:

$$PathCost_{s,d}^{v_1} = ISLCost_{s,v_1} + \sum_{i=1}^{m-1} \lambda_{v_i} ISLCost_{v_i,v_{i+1}} + \lambda_{v_m} ISLCost_{v_m,d}$$ (1)

where $ISLCost$ is the link cost, $\lambda_{v_i}$ is the ISL cost weight factor. When the satellite node $v_i$ is heavily load, increase the $\lambda_{v_i}$, vice versa, decrease the $\lambda_{v_i}$. The goal of load balancing is to find the path $(s,d)$ of minimum $PathCost_{s,d}^{v_1}$.

The goal of QoS routing is to find the path satisfy the delay, bandwidth and packet loss rate requirements. We define path $(s,d)$ should satisfy the three constraints:

$$\text{delay}(path(s,d)) = \sum_{(i,j) \in \text{path}(s,d)} \text{delay}(i,j) \leq D_r$$ (2)

$$\text{band}(path(s,d)) = \min_{(i,j) \in \text{path}(s,d)} (\text{band}(i,j)) \geq B_r$$ (3)

$$\text{loss}(path(s,d)) = 1 - \prod_{(i,j) \in \text{path}(s,d)} (1 - \text{loss}(i,j)) \leq L_r$$ (4)

Where $\text{delay}(path(s,d))$, $\text{band}(path(s,d))$ and $\text{loss}(path(s,d))$ are delay, bandwidth and packet loss rate values of the path $(s,d)$, respectively. The $D_r, B_r$ and $L_r$ are delay, bandwidth and packet loss rate values of the user requirements. The goal of QoS routing is to decrease the average end-to-end delay and packet loss rate, increase the throughput.

ALBQR: Agent-based Load Balancing and QoS Routing

For proposed our ALBQR scheme, we define the path cost based on FARIMA mode and routing table structure. Finally, we give our routing process.

Path Cost Based on FARIMA Model

Before define path cost, we will firstly introduce the ISL cost. The ISL cost can be seen as the sum of propagation delay and queue delay, is defined as:

$$ISLCost = T_{PD} + T_{QD}$$ (5)

where $T_{PD}$ is the propagation delay, $T_{QD}$ is the queue delay. $T_{PD}$ can be calculated by:

$$T_{PD} = \frac{\text{Dist}_{s\rightarrow v}}{V}$$

where $\text{Dist}_{s\rightarrow v}$ is the inter satellite distance, $V$ is the speed of light. $T_{QD}$ can be calculated [7] by:

$$T_{QD} = \frac{1}{\Delta t} \int_{t-\Delta t}^{t} \frac{q(t)}{C} \frac{P_{avg}}{C} dt$$

where $q(t)$ is the occupancy length of the ISL queue at time $t$, $C$ is the ISL capacity, $P_{avg}$ is the mean packet size.
Then, we will define the ISL cost weight factor. We first introduce the
FARIMA model to predict satellite network traffic and get the ISL cost weight
factor. FARIMA model can eliminate the long-range dependent characteristics
existing in the satellite self-similar network traffic [8].

The general form of a FARIMA($p,d,q$) model can be expressed as:

\[ \Phi(z^{-1})(1-z^{-1})^d X_t = \Theta(z^{-1}) \epsilon_t \]  \hspace{1cm} (6)

where \( 0 < |z| < 1 \), \( \Phi(z^{-1}) = 1 - \phi z^{-1} - \phi z^{-2} - \ldots - \phi z^{-p} \), \( p \) order
autoregressive polynomial, \( \Theta(z^{-1}) = 1 - \theta z^{-1} - \theta z^{-2} - \ldots - \theta z^{-q} \), \( q \) order
moving average polynomial, \( \{ \epsilon_t : t = \ldots, -1, 0, 1 \ldots \} \) is a sequence with zero mean
and \( \sigma^2 \) variance. Define \((1-z^{-1})^d\) is the fractional difference operator, 
where \( d \in (-0.5, 0.5) \) is related to the Hurst parameter [9] by the relation of
\( d = H - 0.5 \). Therefore, FARIMA model accurately predict satellite network
traffic.

Suppose the prediction interval set as \( \Delta t \), the congestion level of satellite
node \( \xi(t) \) can be defined as:

\[ \xi(t) = \frac{(Q - q(t)) \times d_{avg} + C \times \Delta t - P}{B} \]  \hspace{1cm} (7)

where \( Q \) is length of queue, \( q(t) \) is the occupation length of queue, \( d_{avg} \) is the
average packet size, \( P \) is predicted traffic value at time \( t + \Delta t \), \( C \) is the ISL
capacity, \( B \) is size of buffer. The ISL cost weight factor \( \lambda \) can be calculated as:

\[ \lambda = \frac{1}{1 + \exp(-a \xi)} \]  \hspace{1cm} (8)

where \( a \) is slope parameter. Finally, we get the \( PathCost_{s,d}^{v_i} \) of \( path(s,d) \) as
formula (1).

Routing Table

ALBQR scheme routing table is a probabilistic routing table, which is the
probability value of selecting neighbor satellite nodes for different destination
nodes. Suppose the satellite \( s \) deliver packets to destination satellite \( d \) via the
neighboring satellite \( j \), the probability satisfies: \( \sum_{j \in N(s)} p_{s,d}^j = 1 \), where \( N(s) \) is
the neighboring set of satellite \( s \). The TABLE I shows the routing items of satellite \( s \) to destination satellite node \( d \).

| destination | neighbor | probability |
|-------------|----------|-------------|
| d           | \( v_1 \) | 0.2         |
|             | \( v_2 \) | 0.4         |
|             | \( v_3 \) | 0.1         |
|             | \( v_4 \) | 0.3         |
The routing table can be initialized according to the following equation:

\[
p_{s,d}^j = \begin{cases} 
\frac{1}{|N(s)|} & j, d \in N(s) \land j = d \\
\frac{4}{3} \times \frac{|N(s)|-1}{|N(s)|^2} & j, d \in N(s) \land j \neq d \\
\frac{1}{|N(s)|} & j \in N(s) \land d \not\in N(s)
\end{cases}
\]  

(9)

Then, the path cost updating as following equation:

\[
PathCost_{s,d}(t) = \gamma PathCost_{s,d}(t - 1) + (1 - \gamma) PathCost_{s,d}^{v_i}(t)
\]

(10)

where PathCost_{s,d}(t - 1) is the path cost of path(s, d) at time t - 1. PathCost_{s,d}^{v_i}(t) is the path cost path(s, d) via node v_i at time t, the \( \gamma \) is the weight of PathCost_{s,d}(t - 1), \( \gamma \in [0,1] \). Finally, the routing table updating as following equation:

\[
p_{s,d}^j = \begin{cases} 
\frac{p_{s,d}^j + \alpha \Delta Q}{1 + \alpha \Delta Q} & j, v_i \in N(s) \land j = v_i \\
\frac{p_{s,d}^j}{1 + \alpha \Delta Q} & j, v_i \in N(s) \land j \neq v_i
\end{cases}
\]  

(11)

\[
\Delta Q = e^{-\beta \times PathCost_{s,d}^{v_i}(t) / PathCost_{s,d}(t)}
\]

(12)

where \( \alpha \) and \( \beta \) are the real number and \( \alpha, \beta > 1 \). \( \Delta Q \) is the new path influence factor.

**Routing Process of ALBQR**

The ALBQR routing process can be described as follows:

**Step1** For all satellites, stationary agents initialize routing tables according formula (9). From each satellite s, a normal agent migrates towards a destination satellite d to explore low-cost path. And destination satellite is randomly selected.

**Step2** When the source satellite s receive the service request with QoS parameter, the source satellite s send the forward QoS mobile agents to the destination satellite node d with a maximal number of hops \( max_{hops} \) and time-to-live.

**Step3** Forward agents uses high-priority queues to quickly explore candidate QoS routing path. On the intermediate satellite v choose next hop as follows:

- If the forward agent number of hops is greater than \( max_{hops} \) or reaches the max time-to-live, remove this forward agent.
- If the forward agent all neighbors have been visited, remove this forward agent.
- The next hop must satisfy QoS constraints. If there is no inter-satellite link that can satisfy the minimum bandwidth requirement, remove this forward agent. If the total delay \( ISLCost_{s,v_1} + \sum_{i=1}^{m-1} ISLCost_{v_i,v_{i+1}} > \)
\textbf{Dichromatic}, remove this forward agent. If the total packet loss rate \(1 - \prod_{(i,j) \in \text{path}(s,d)} (1 - \text{loss}(i,j)) > L_r\), remove this forward agent.

- Then, we get all the not yet visited neighboring nodes and satisfy the QoS constraints node, and the next hop is selected based on the roulette wheel strategy and the following formula:

\[
p_{s,d}^{j} = \frac{p_{s,d}^{j}}{\sum_{l \in N'(s)} p_{s,d}^{l}}
\]  

(13)

where \(N'(s)\) is the reachable neighboring node of satellite \(s\). We normalized the routing table probability \(p_{s,d}^{j}\) and then operate roulette wheel strategy to select the next hop.

**Step 4** While the forward agents migrating to the destination satellite node, they gather the ISL cost, packet loss rate of intermediate satellite node \(v\), and push these information to stack.

**Step 5** When the forward agents reach the destination satellite node, we get the satisfy load balancing and QoS path, and then the forward agents transform to backward agents. The backward agents use high-priority queue migrate back to source node in opposite direction and pop its stack information to get next hop.

**Step 6** When the backward agents migrate to intermediate satellite \(v\), the backward agent interacts with stationary agents and notifies stationary agents reserve bandwidth resource. Stationary agents updating path cost by formula (11) and updating routing table by formula (12).

**Step 7** If no backward agents migrate back to source satellite node at a time interval, the service request is rejected. If only a backward agent migrates back to source satellite node, we select this return path. If multiple backward agents migrate back to source satellite node, then we select the routing path according the following selecting strategy:

Suppose from satellite node \(s\) to satellite node \(d\) have \(k\) routing path that satisfy QoS requirements, \(P = (p_1, p_2, ..., p_k)\). We define cost function as:

\[
C_i = \lambda \times \frac{\text{TotalDelay}(p_i)}{\sum_{l=1}^{k} \text{TotalDelay}(p_l)} + \mu \times \frac{\text{TotalLoss}(p_i)}{\sum_{l=1}^{k} \text{TotalLoss}(p_l)}
\]  

(14)

where \(\lambda\) and \(\mu\) are delay and packet loss rate weight, \(\lambda \in [0,1]\), \(\mu \in [0,1]\) and \(\lambda + \mu = 1\).

**PERFORMANCE EVALUATION**

In this paper, we use OPNET14.5 to simulate the proposed routing scheme. We use ELB (Explicit Load Balancing) and DSP (Dijkstra Short Path) [4] as comparison routing schemes, and we set \(\alpha = 3, \beta = 5, \gamma = 0.7, \lambda = 0.6, \mu = 0.4\). We simulated Iridium satellite constellation, the simulation parameters are presented in TABLE II.
TABLE II. Simulation Parameters.

| Parameters                  | Value                                           |
|-----------------------------|-------------------------------------------------|
| Number of planes            | 6                                               |
| Number of satellites per plane | 11                                              |
| Orbit altitude              | 778km                                           |
| Orbit inclination           | $86.4^{\circ}$                                  |
| Max number of ISLs          | 4 (2 Intra-plane + 2 inter-plane)               |
| Polar region boundary latitude | $\pm 70^{\circ}$                               |
| Min elevation angle         | 10$^{\circ}$                                    |
| ISL queue Type              | FIFO                                            |
| ISL queue length            | 500Packets                                      |
| ISL bandwidth               | 10Mb/s                                          |
| UDL bandwidth               | 15Mb/s                                          |
| Delay requirement           | 200ms                                           |
| Bandwidth requirement       | 200kb/s                                         |
| Packet loss rate requirement| 0.04                                            |

We use the following indicators to estimate the performance of ALBQR: traffic prediction error rate, normalized ISL load, network throughput, average packet loss rate, and average end-to-end delay.

Normalized Mean Square Error (NMSE) was used to estimate the traffic prediction error rate. The NMSE equation as follows:

$$NMSE = \frac{\sum_{i=1}^{N}(\hat{Y}(i) - Y(i))^2}{N} / \frac{\sum_{i=1}^{N} Y(i)^2}{N}$$  \hspace{1cm} (15)

where $Y(i)$ is real traffic, $\hat{Y}(i)$ is predict traffic. Through the simulation, we can get the FARIMA simulation result as Figure 2.

Figure 2. The prediction performance of FARIMA.
From the Figure 2, we can see the NMSE of FARIMA is $1.338e^{-4}$, the FARIMA model can accurately the satellite traffic.

For estimate the performance of ALBQR load balancing, let we use the Normalized ISL load [10]:

$$Normalized\ ISL\ Load = \frac{Accumulated\ Traffic\ Volume(Mb)}{Time\ Period(s) \times Total\ Link\ Capacity(Mb/s)}$$

Normalized ISL load is accumulated traffic volume divide by inter-satellite link bandwidth, this value is never exceeded 1.

The change of normalized ISL load of a satellite running from south region to north region showed as Figure 3. We set the terminal bitrate is 500kb/s, the normalized ISL load peak value merges in latitude 30°. ALBQR exhibits much lower in high traffic region and much smoother than ELB and DSP, which demonstrate ALBQR have better ability of load balancing.

![Figure 3. Normalized ISL load for different latitude.](image)

![Figure 4. Throughput for different terminal bitrate.](image)
Figure 4 graphs the throughput versus terminal’s bitrate. ALBQR achieves better throughput compare with ELB and DSP. Especially, when bitrates are higher than 600kb/s, ALBQR’s throughput is increased by approximately 10% than ELB and DSP.

Figure 5 depicts the average end-to-end delay for different terminal bitrate. ALBQR’s average end-to-end delay is longer than ELB and DSP, when terminal bitrate below 700kb/s. The reason is ALBQR may goes through more hops to get the QoS path and avoid congestion, ELB and DSP may get the shortest path. However, ALBQR’s average end-to-end delay increase smoother and lower than ELB and DSP in the case of high traffic.

Figure 5. Average end-to-end delay for different terminal bitrate.

Figure 6. Mean packet loss rate for different terminal bitrate.
Figure 6 shows the mean packet loss rate for different terminal bitrate. When the terminal’s bitrate below 700kb/s, the mean packet loss rate of ALBQR, ELB and DSP are all close to 0, this is due to the network is idle. However, ALBQR’s mean packet loss rate is much lower than ELB and DSP in the case of high terminal bitrate. This is due to ALBQR avoid the heavy load satellite nodes.

CONCLUSION

In this paper, we propose ALBQR, an agent-based load balancing and QoS routing scheme for the LEO satellite network. ALBQR dispatches mobile agents migrating from source satellite towards destination satellite to explore load balancing and QoS path. The Iridium satellite network performance has been evaluated in terms of normalized ISL load, throughput, average end-to-end delay and mean packet loss rate. ALBQR can achieve better normalized ISL load and throughput, and lower mean pack loss rate. ALBQR can achieve lower end-to-end delay in case of heavily load. Future work will include designing more accurately traffic prediction model or other ISL cost modification factor.

ACKNOWLEDGEMENT

The research was supported by Fund of State Key Laboratory of Information Photonics and Optical Communications (Beijing University of Posts and Telecommunications), P. R. China.

REFERENCES

1. Liu X, Xuanwen M. A Self-adapting Traffic Class Routing in LEO/MEO Satellite Networks[J]. Journal of Convergence Information Technology, 2011, 6(10):155-163.
2. Liu Z, Li J, Wang Y, et al. HGL: A hybrid global-local load balancing routing scheme for the Internet of Things through satellite networks[J]. International Journal of Distributed Sensor Networks, 2017, 13(3):155014771769258.
3. Wu Z, Hu G, Jin F, et al. A novel routing design in the IP - based GEO/LEO hybrid satellite networks[J]. International Journal of Satellite Communications & Networking, 2017, 35(3):179-199.
4. Huang J, Liu W, Su Y, et al. Load Balancing Strategy and Its Lookup-table Enhancement in Deterministic Space Delay/Disruption Tolerant Networks[J]. Advances in Space Research, 2017, 61.
5. Rao Y, Zhu J, Yuan C A, et al. Agent-based Multi-Service Routing for Polar-orbit LEO broadband satellite networks[J]. Ad Hoc Networks, 2014, 13(1):575-597.
6. Song G, Chao M, Yang B, et al. TLR: A Traffic-Light-Based Intelligent Routing Strategy for NGE0 Satellite IP Networks[J]. IEEE Transactions on Wireless Communications, 2014, 13(6):3380-3393.
7. Rao Y, Wang R. Performance of QoS routing using genetic algorithm for Polar-orbit LEO satellite networks[J]. AEU - International Journal of Electronics and Communications, 2011, 65(6):530-538.
8. Das J, Banerjee R. Estimation of long range correlations and FARIMA modelling of wind speed in Maharashtra[C] IEEE Pes Asia-Pacific Power and Energy Engineering Conference. 2017:1-6.
9. SHENG H, CHEN Y Q. On the robustness of Hurst estimators[J]. Signal Processing, IET, 2011(5): 209-225.
10. Rao Y, Wang R C. Agent-based load balancing routing for LEO satellite networks[J]. Computer Networks, 2010, 54(17):3187-3195.