A Robust Routing Strategy based on Deep Reinforcement Learning for Mege Satellite Constellations

Ke Chu, Sixi Cheng and Ying Yang

National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, 611731, Chengdu, China

The development of mega constellations inevitably brings various problems for the development of routing techniques. Most of the existing work considers end-to-end delay and load balancing problems, while the analysis of routing strategies in case of link performance degradation is neglected, and an optimization approach applicable to mega satellite networks is not developed. In this letter, we propose a robust routing strategy based on deep reinforcement learning (RRS-DRL) that regards the Age of Information (AoI) of packets as an optimization target, and ensures the effectiveness of message transmission throughout the network. Extensive simulation results show that our proposed RRS-DRL algorithm obtains a lower average AoI across the network and better utilization of the resources than the traditional shortest path algorithm, significantly increasing the robustness of the constellation.

Introduction: In recent years, the explosive growth of the number of LEO (low-Earth orbit) satellites has brought many opportunities and challenges to the development of non-terrestrial network [1], which use mega satellite constellations(e.g. Starlink, OneWeb and Kuiper). The mege satellite networks face many challenges, the most important of which is the security of routing. Once the satellite network routing is attacked by malicious network behavior, it will likely to be paralyzed. At the same time, an inter-satellite routing technique with robustness for mega satellite constellations has become a hot research topic. Due to the increasing number of users and the potential for jamming or performance degradation, routing technology is facing the problem of the increasing load and topology changes. The size of routing table is growing at an extremely fast rate, resulting in a waste of satellite Internet systems resources.

How to route in satellite networks is a NP-hard problem [2]. In the traditional satellite Internet routing algorithm, it is obvious that the centralised offline preset strategy is difficult to apply to the dynamic unknown environments. The intelligent methods have more applications when dealing with various threats and unexpected situations. Their advantage becomes more obvious as the complexity of the problem becomes higher and the size of the data in the problem becomes larger.

Since reinforcement learning can explore the environment through trial and error, it has been widely used in routing algorithms. [3] proposed a satellite routing algorithm based on reinforcement learning to find the optimal transmission path, and also improved the traditional algorithm's problem of long convergence time and sometimes falling into local convergence by limiting the number of hops of the algorithm and adopting dynamic greedy coefficients. However, the text assumes that the network topology is static and free from being jammed, which is clearly unrealistic. Some studies [4] proposed deep reinforcement learning methods for solving global routing problems in simulated environments with the reward from many aspects, but is simply used for the problem of finding path, without considering the load balancing problem arising from the degradation of link performance when subjected to continuous disturbances, and the consideration of a single problem make it hard to migrate the model. Regarding the consideration of more reward functions, some studies [5] propose energy-efficient routing protocols based on deep reinforcement learning, and again the study Lack of consideration for jamming. These methods are less able to generalise to more features and are less responsive to environmental changes. Other researchers have proposed a proposed Fast Response Anti-Jamming Algorithm (FRA) [6] with the goal of minimising the anti-jamming routing overhead and investigated anti-jamming routing schemes for heterogeneous internet of satellite (IoS), but the authors separated the process of path-finding and anti-jamming, making the complexity increases.

For scenarios where jamming is considered, we propose a robust routing strategy based on deep reinforcement learning (RRS-DRL). We use a priori information about the jamming as part of the state, enabling the routing algorithm to deal with dynamic network changes. Secondly, as it is also crucial for meeting timely responses to emergency events. We combine the Age of information (AoI) of packets to reward and punish the selection of the next hop, ensuring the effectiveness of messages collection and transmission across the network and realising the time value of information. In this letter, a mega, dynamically changing satellite network is used to analyse routing performance, and we consider not only link failures but also link performance degradation, allowing RRS-DRL to be aware of jamming through trial and error. Simulation results show that RRS-DRL has low delay jitter and low average information age across the network, effectively improving the robustness of the satellite constellation network.

System Model: This section describes the LEO satellite network model, the packet transmission model and the problem of anti-jamming routing.

A. Network Mode.

The structure of a LEO satellite network can be represented as an abstract diagram $G=\{\eta,\mathcal{L}\}$, where η represents the set of satellites, ζ represents the set of links between satellites, and each edge e_{ij} represents the weight between V_i and V_j.$e_{ij}=\lambda_{ij}^k[b]\cdot\psi_{ij}^k$, ψ_{ij}^k represents the propagation delay on the link (i,j) when the kth packet arrives. $\lambda_{ij}^k[b]$ is an indicator of link connectivity and $\lambda_{ij}^k[b]$ is expressed as follows:

$$\lambda_{ij}^k[b]=\begin{cases}1 & \text{if th link}(i,j)\text{ is activated in channel b}\\ & \text{for kth packet}\\ 0 & otherwise\end{cases} \tag{1}$$

To model the jamming between links, we denote $\mathcal{L}i$ as the set of links in $\mathcal{L}$ that are within the jamming range at the time of t_i . Then, for the nodes in it we have:

$$\sum_{l\in\mathcal{L}i,t\in t_i}\lambda_{ij}^k[b]+\sum_{l\in\mathcal{L}i,t\in t_j}\lambda_{ij}^k[b]\leq 1, \tag{2}$$

where $b\in\mathcal{B}$. This means that if a packet k is received on channel b, it will not be interfered with on the same channel by an unintended transmitter p in the t_j jamming range.

B. Packet Transmission Model.

Assume that the sender of all sessions is continuously updated and the generated data is split or reorganized into packets of uniform size for transmission, the length of the packet is noted as d. ν^l is the generation rate of packet k at the source s_k, the packet generation interval at the sender s_l is a constant $1/\nu^l$. C_{ij} indicates the link capacity in the link (i,j) with:

$$C_{ij}=W_{ij}\log_2\left(1+\frac{(p_i{f_i}^2)/{d_{ij}}^2}{n_0W_B(4\pi/c)+(p_\chi{f_\chi}^2)/{d_{ij}}^2}\right), \tag{3}$$

where W_{ij} and c are the channel bandwidth and speed of light, respectively. p_i,p_χ,f_i,f_χ are the power and frequency from the transmitting node i and the jammer respectively, d_{ij} is the distance between the nodes i and j. And n_0 is the ambient Gaussian noise density. When the link is interfered with, the signal-to-noise ratio(SNR) is reduced, thus affecting the link capacity. Let μ_{ij} denote the transmission rate of the link (i,j) and the rate is limited by the total capacity of the link:

$$\mu_{ij}\leq C_{ij}\cdot\sum_{b\in\mathcal{B}}\lambda_{ij}^k[b]. \tag{4}$$

When the network reaches a steady state, in order to avoid packet loss due to an infinite number of backlogged messages at any relay node, the transmission time for each message on the link should not be greater than the time interval between messages generated by the session using the message. Therefore, each walk able link has $1/\nu^l\geq d/\mu_{ij}$.

In traditional path planning problems, the metrics adopted are relatively homogeneous.A newly introduced metric to measure the timeliness of information updates, AoI [7], is used in our paper. It is calculated as follows:

$$\begin{aligned} A_a^l &= \frac{1}{T}\sum_{k=1}^{K}\left(\frac{1}{2\nu^{l2}}+\frac{d}{\nu^l\mu_{sla}}+\frac{d_{sl\_to\_a}}{\nu^l c}\right) \\ &= \frac{K}{T}\frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{2\nu^{l2}}+\frac{d}{\nu^l\mu_{sla}}+\frac{d_{sl\_to\_a}}{\nu^l c}\right) \\ &= \nu^l\left(\frac{1}{2\nu^{l2}}+\frac{d}{\nu^l\mu_{sla}}+\frac{d_{sl\_to\_a}}{\nu^l c}\right) \\ &= \frac{1}{2\nu^l}+\frac{d}{\mu_{sla}}+\frac{d_{sl\_to\_a}}{c} \end{aligned}. \tag{5}$$

Point a is the destination node, $\mu_{s_l a}$ denotes the transmission rate from node s_l to node a. It can be deduced that the AoI at the next node of the current satellite node is:

$$A_{d_l} = \frac{1}{2\nu^l} + \sum_{i\neq d_l,\lambda_{ij}^l=1}\left(\frac{d}{\mu_{s_l a}}+\frac{d_{i\_to\_j}}{c}\right). \tag{6}$$

The time-averaged AoI can be calculated by:

$$\begin{aligned} A_{ave} &= \sum_{l\in\mathcal{L}} A_{d_l} = \sum_{l\in\mathcal{L}}\left(\frac{1}{2\nu^l}+\sum_{i\neq d_l,\lambda_{ij}^l=1}\left(\frac{d}{\mu_{ij}}+\frac{d_{i\_to\_j}}{c}\right)\right) \\ &= \sum_{l\in\mathcal{L}}\frac{1}{2\nu^l}+\sum_{i\in\eta}\sum_{j\in\mathcal{T}_i,\lambda_{ij}^l=1}\left(\frac{d}{\mu_{ij}}+\frac{d_{i\_to\_j}}{c}\right) \end{aligned} \tag{7}$$

$\mathcal{T}_i$ Indicates each satellite in the path node set $\mathcal{T}_i$, that session l passes through from a node satellite η.

C. Anti-jamming Routing Strategy Model

Specifically, in this paper, for packets of a given source and destination, our goal is ensure the reliability of the link with an jammed scenario, while guaranteeing the average information age of all sessions. Also, since satellite node caches all have a certain queue length, the length of packets entering the queue cannot exceed the maximum transmission capacity of each node.Based on the above analysis, the anti-jamming routing problem with guaranteed delays can be formalised as:

$$\begin{aligned} &\mathrm{OPT}\min\left(A_{\mathrm{ave}}\right) \\ &\text{s.t.Jamming} \quad \text{constraints}:(2); \\ &Packet \quad Transmission \quad Model:(4); \\ &\text{The total time averaged AoI function}:(7). \\ &Pkt_{\Delta_T} \leq Pk_{\max}^{rev}+Pk_{\max}^{send} \end{aligned}$$

The RRS-DRL Algorithm.: In this section, we propose a robust routing algorithm based on deep Q-network(DQN) to obtain the perception of satellite link delay with respect to the age of node information without precise knowledge of the propagation delay. Once the performance of some satellites degrades or the corresponding links break, packets are routed to satellites with better performance and higher capacity.And this algorithm is applicable in constellations with mega LEO satellite and high state dimension, which provides ideas for robust routing strategies for complex networks.

A. Routing Agent Design

For packets, the node selection in the given link depends only on the node selection in the previous link. The action taken by each packet is only related to the current state of the network.It can typically be described as a Markov Decision Process (MDP) [8].

The key elements of the MDP are given below:

i): Status S: $(Node_{currpos}, Node_{dest}, \chi_{t_i})$, $Node_{currpos}$ indicates the number of the current node, $Node_{dest}$ is the number of the destination ,χ_{t_i} the form of jamming received, which provides sufficient information to learn the best strategy for routing.

ii): Action A: $Node_{next}$, the next-hop neighbour node reachable by the satellite node(generated according to the network topology).

iii):Reward r: The reward is defined as a function $R(a, s')$, a denotes the selected action, and s' denotes the next node number reached after the node selection action. Traditional routing algorithms based on reinforcement learning do not work well for routing in LEO satellite constellations because it is difficult to iterate due to the large state dimension. In most cases, it is assumed that links will not fail or that nodes will be directly unavailable after jamming, but in practice, in LEO constellations, not only jamming avoidance but also load balancing is required using routing strategies when sweeping jamming is encountered and when nodes are partially available but with degraded performance. To overcome the problem of jamming in the network, we add very few priori knowledge of jamming to the state to enhance the learning ability of the agent to the environment. The following elements are used to set up the reward function:

The distance between the next satellite node and the destination node Dis_i. To train the agent to know how to send packets to the destination without passing through redundant satellite nodes, we should add a distance penalty so that the agent automatically chooses the shorter path.

AoI of the packet which arrives at the next hop $AoI_{s'}$. Aol is designed to quantify time-critical updates at the receiving end, characterising the timely delivery of information at the destination, and increasing the Aol penalty ensures that information is fresh.

The queue growth rate $\vartheta_{s'}$, which sets a penalty proportional to the queue growth rate when the queue size exceeds the threshold, is set to reduce packet buildup and increase the probability that traffic will be distributed evenly.We define the reward as a weighted sum of these factors:

$$\mathrm{r} = w_1 * Dis_i + w_2 * AoI_{s'} + w_3 * \vartheta_{s'} \tag{8}$$

The routing policy is the probability distribution of all actions in each state, denoted by $\pi(s|a)$ The goal of the proposed routing MDP is to find the policy that maximizes the reward for discount accumulation $\pi*$.

$$P:\max_{\pi^*}\left(\sum_{\tau=0}^{\infty}\gamma^\tau r_{t+\tau} \mid s_t = s, a_t = a\right) \tag{9}$$

$\gamma \in [0, 1)$ is the discount factor,the weight between immediate and subsequent rewards. The next section will use RRS-DRL algorithm to solve the above MDP problem.

B. RRS-DRL Algorithm

In recent years, DRL has been widely used to deal with sequential decision problems with high-dimensional states. DQN [9], a typical algorithm in this context, is based on the principle of using a deep neural network to fit Q-values. The input state s includes the current node of the packet, the destination node, and the current topology state number. With this information as input. DQN uses the approximate action value function $Q(s, a, \theta)$ from the network output to approximate the actual action value function $Q_\pi(s, a)$, where θ is the neural network parameter. The actions are selected by Q function $\mathrm{Qfunction\,a} = \arg\max Q\left(s, a', \theta\right)$. If the next node meets the packet restriction condition, the packet is forwarded. Execution of action a obtains the corresponding reward r and the new state s'. The transitions in the memory replay pool are then stored $e = (s, a, r, s')$.The reward value includes the distance to the end point after , the age of the message, and the queue growth rate fed by the next node. To update the Q network, the agent selects a random batch of transitions and calculates the loss.

$$L(\theta) = E_{\pi_\theta}\left[\sum\left(r+\gamma * \mathcal{V}\left(Q^*\left(s', a, \theta^-\right)\right) - Q(s, a, \theta)\right)^2\right] \tag{10}$$

Q^* is the target q-value and θ^- is the parameter for the target q-value. The parameters are updated using the stochastic gradient descent (SGD) method gradient,and the gradient is calculated as:

$$\nabla_\theta L(\theta) \approx E_e\left[\left(r+\gamma * \mathcal{V}\left(Q^*\left(s', a, \theta^-\right)\right) - Q(s, a, \theta)\right)\right] \tag{11}$$

where $\alpha > 0$ is the learning rate.

To enhance the 'exploration' of the environment, the dynamic ε-greedy algorithm is considered in the early stages of training. The value ε is set as a decreasing dynamic value from ε_0 to ε_1, depending on the number of iterations, and is denoted as:

$$\varepsilon = \varepsilon_0 \cdot \varepsilon_f^i$$

where i is the number of iterations, ε_0 is the initial value, and ε_f is the decay rate.

As shown in the Fig.1, we train the routing strategy using the state information (current node, destination node, and topology state

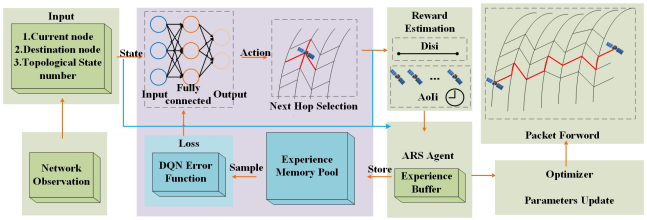


Fig. 1. RRS-DRL Framework

number) in the satellite network. the Agent observes the state from the environment (i.e., satellite communication network) and selects the next node from the accessible satellite nodes as the action , the The episodic interaction ends once all the packets reach the destination node. And we obtain the AoI of all packets at the destination node. The structure of the deep neural network is also shown in Fig.1. Usually, it contains two fully connected layers (of size 175) for extracting the features of the satellite network. The output features are sequentially input to the output layer to obtain the final value function.

Every time a packet arrives at a new node, the satellite communication network updates its internal state, feeding back a reward r,an experience term (s_t, a_t, r_t, s_{t+1}) is stored in the experience pool, and the model parameters are updated with the sampled values therein.

Algorithm RRS-DRL Algorithm

Initialize: Learning rate α, discount factor γ, network weight θ, episode memory $\mathbb{E}$, memory replay pool $\mathcal{D} = \emptyset$
for episode = 1 to ∞ **do**
  for step= 1 to $step - num$ **do**
    Sence env and abtain s_t
    Select a random action a according to $\varepsilon - greedy$,
    Excute a_t
    if $Pkt_{\Delta_T} \leq Pk_{\max}^{rev} + Pk_{\max}^{send}$ **then**
      Forward packet as action a_t
      abtain reward r_t and new state s'
    end if
    Store transition (s_t, a_t, r_t, s_{t+1}) in D
    Sample random minibatch of transitions (s_t, a_t, r_t, s_{t+1}) from D
    Update θ
    Calculate gradient by Equa.11
    Update network every C steps
  end for
end for

Numerical Results and Discussions.: The parameters in this paper are taken from Starlink's FCC file, and the bilinear element set coordinates of Starlink's satellites in different orbits are also available, from which the distance of the inter-satellite link can be calculated. During the simulation, 175 satellites were selected as experiments.

Track height	550km
Minimum angle of elevation	25°
Number of tracks	7
Number of satellites per orbit	25
Inclination	53°

Table 1: Satellite Network Parameters.

We built the network using Python and networkX packages and evaluated the performance of the RRS-DRL in terms of average message age and delay jitter across the network. The hyper-parameters are set as follows: discount factor γ is 0.6, ε_0 is 0.7, ε_f is 0.975, learning rate is 0.005, memory batch size is 16, and size of memory replay pool $\mathcal{N}$ is 1000. The target network update step is 10, the activation function is Tanh.

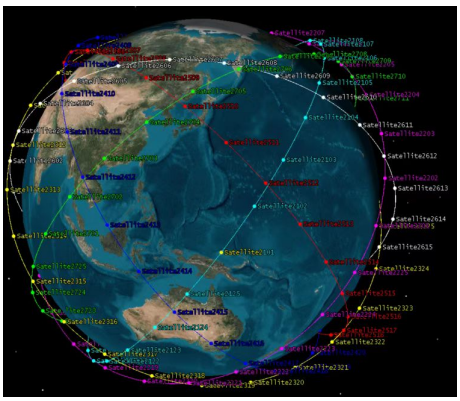
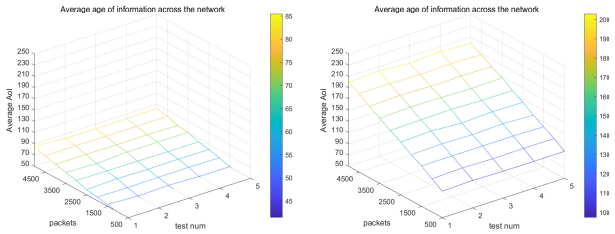


Fig. 2. Satellite Network

In the simulations, the presence of sweeping jamming in the environment causes the link performance degrade. FRRSt, many packets are generated on the network (network load), each with a random source and destination node. Each time a packet is transmitted, a new packet is initialised after a certain time step. Once a certain number of packets have been generated and transmitted on the network, the simulation ends. We use the average AoI and delay jitter values as performance evaluation metrics. To evaluate the effectiveness of our approach, we compare the traditional routing algorithms: shortest path fRRSt (SPF) [10]. We then present the evaluation results and analysis.

A.Average AoI

In this letter, the two algorithms are simulated under the load of 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000 packets, and multiple rounds of tests are conducted under each load. As shown in the Fig.3 and , it is found that the RRS-DRL algorithm can still maintain relatively low delay with the increase of load, and the results are stable, showing strong adaptability to the network. Although, the SPF algorithm performs well when the number of loads is less than 2500 packets, as the number of loads continues to increase, the age of the full network information is higher compared to the RRS algorithm. This is because SPF algorithm has poor adaptability to the network, and it is more difficult to adjust the impact of jamming when the number of loads increases. At this time, the performance of SPF degradation was found.



((a)) RRS-DRL-AOI. ((b)) SPF-AoI.

Fig. 3. Average AoI.

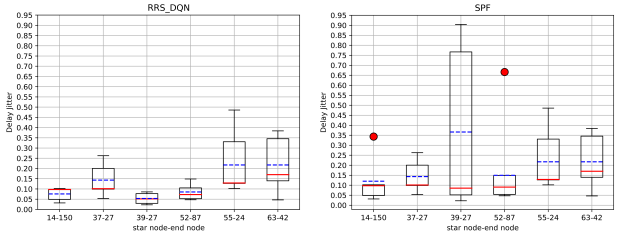
B.Time Delay Jitter

The jitter is the variation of network delay, which is generated by any two adjacent packets of the same application (the same start node and destination node) passing through the network delay in the transmission route. It is obtained by dividing the delay time difference of adjacent packets by the difference of packet sequence number, or by the information age difference of adjacent packets. Time delay jitter can be calculated as follows:

$$\tau_{[sl,dl]} = \frac{AoI_j - AoI_i}{j - i} \tag{12}$$

We took several of these paths for analysis of the delay jitter values in the simulation, as shown in box line Fig.4, the red line represents the median of the delay jitter, the upper and lower bound

of the box represents the upper and lower quartile, we can also see the maximum and minimum values from the Fig.4, we can see that in the RRS-DRL algorithm, there are no data outliers, while the SPF algorithm not only has delay outliers, and the higher the delay jitter. With lower levels of delay jitter, the RRS-DRL is more resilient to dynamic changes in the network, not only in the case of network failure, but also in the case of node performance degradation.



((a)) RRS-DRL-DJR. ((b)) SPF-DJR.

Fig. 4. Time Delay Jitter Rate

Conclusion: For the problem of routing large networks in the existence of jamming, a deep reinforcement learning-based robust routing strategy is proposed in this letter. We consider the situation that not only the link is damaged, but also some link performance is degraded when it is disturbed. Meanwhile, the age of satellite node information is incorporated into the reward function as the optimization objective to ensure the transmission effectiveness of the whole network. According to the simulation results, the method obtains a lower average information age of the whole network and a lower delay jitter rate compared to SPF, which increases the robustness of the constellation.

Acknowledgment: This work is fully supported by Natural Science Foundation of China Project (61871422).

References

1 O. Kodheli et al., "Satellite communications in the new space era:A survey and future challenges," *IEEE Commun. Surveys Tuts*, vol. 23, no. 1, pp. 70–109, 1st Quart. 2021.
2 R. Minimum et al., "Minimum flow maximum residual routing in LEO satellite networks using routing set," *Wirel. Netw.*, vol. 144, no. 4, pp. 501-517, Aug. 2018.
3 Wang, X. and Dai, Z. and Xu, Z., "Minimum flow maximum residual routing in LEO satellite networks using routing set," *Wirel. Netw.*, pp. 1105-1109, May 2018.
4 Liao, H. and Zhang, W et al., "A Deep Reinforcement Learning Approach for Global Routing," *J MECH DESIGN*, vol. 142, no. 6, pp. 1-17, Jun. 2019.
5 Liu, J and Zhao, B et al., "DRL-ER: An Intelligent Energy-Aware Routing Protocol With Guaranteed Delay Bounds in Satellite Mega-Constellations," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 4, pp. 2872-2884, Dec. 2021.
6 Han, C. and Liu, A et al., "Anti-Jamming Routing For Internet of Satellites: a Reinforcement Learning Approach," *ICASSP*, pp. 2877-2881, Barcelona, Spain, 2020.
7 Lou, J. and Yuan, X et al., "AoI and Throughput Tradeoffs in Routing-aware Multi-hop Wireless Networks," *IEEE. INFOCOM 2020*, pp. 476-485, Toronto, ON, Canada, 2020.
8 Sutton, R. S. and Barto, A. G., "Reinforcement Learning," *IEEE. INFOCOM 2020*,vol. 15, no. 7, pp. 665-685, 1998.
9 V. Mnih, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529-533, Jan. 2015.
10 Dijkstra, E. W., "A note on two problems in connexion with graphs," *Numer Math (Heidelb)*,vol. 1, no. 1, pp. 269–271, Dec, 1959.