An Edge Computing-Based Framework for Marine Fishery Vessels Monitoring Systems

Fengwei Zhu[1], Yongjian Ren[1], Jie Huang[1](✉), Jian Wan[1], and Hong Zhang[2]

[1] School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China
huangjie@hdu.edu.cn
[2] Institute of Science and Technology Information Research of Zhejiang Province, Hangzhou 310006, China

Abstract. Vessel Monitoring Systems (VMS) have been adopted by many countries which provide information on the spatial and temporal distribution of fishing activity. Real-time communication and interaction between fishing vessels and shore-based systems is a weakness of traditional vessel monitoring systems. This paper proposes a novel framework of edge computing-based VMS (EC-VMS). The framework of EC-VMS mainly consists of four layers. An edge computing terminal is used on each vessel, and the BeiDou navigation satellite system (BDS) is adopted for communication. Meanwhile, edge computing servers interact with corresponding management vessels and the cloud. In order to decrease the communication cost, a data transmission policy called Adaptable Trajectory Transmission Model (ATTM) is presented in this paper. The experimental results illustrate the efficiency of the proposed EC-VMS, with the average communication time significantly decreased in a typical scenario. Moreover, EC-VMS improves the real-time performance of the system.

Keywords: VMS · Edge computing · BDS · Marine fishery

1 Introduction

Currently Vessel Monitoring Systems (VMS) are widely adopted by many countries around the world to allow fisheries administrators to control and monitor fishing activity. The electronic modules are installed on-board vessels which can automatically send data to a base station on shore by satellite communication. The fisheries monitoring center receives the transmitted data and processes it to get vessel trajectories and other information. Utilizing information on the vessels near-real time location, along with the vessel movements information that the VMS gives many benefits such as improving the quantity and quality of logbooks recovered, obtaining access to fishery-independent fishing effort estimates and prompt catch/effort re-porting, enabling the possibility of regional management and understanding both fleet dynamics and vessel behavior, and increasing efficiency of vessel safety protection [1].

X. Wang et al. (Eds.): CollaborateCom 2019, LNICST 292, pp. 201–214, 2019.
https://doi.org/10.1007/978-3-030-30146-0_14

Nevertheless, VMS still has some shortcomings in real-time and maritime communications. For instance;

(1) The development of marine communication networks is much slower than that on land, marine communication systems available today only provide the bare minimum essential services such as ship identification, positioning, location, course, heading, destination, tonnage and speed etc. This is provided in the form of AIS (Automatic Identification System) using VHF radio frequencies. Inter ship satellite communication is possible but is a costly option when compared to conventional wireless communications and not affordable for most small to medium seagoing vessels [2]. Sensor devices deployed on vessels can generate huge volumes of useful data that require significant portions of bandwidth for dissemination but it not utilized due to the deficiencies of the communication network.
(2) Fishing activity is monitored to detect vessels committing infringements, which requires near real time information dissemination so that the suspected infringements can be immediately detected. The processing of such data in the cloud faces additional delay due to wide area network latency that hinders the real-time response [3].

To address the problem, vessel monitoring systems are adopting more intelligent technologies to manage all the vessels. This paper proposes an edge computing-based intelligent VMS (EC-VMS) for smart vessel management. Every vessel has a perception platform to interact with the vessel terminal, sensors and other condition data collectors. Therefore, it can monitor itself in real-time and provide the data to the server. As the BeiDou navigation satellite system (BDS) can be used for positioning and communication through short messaging, the EC-VMS adopts it for communication. Thus, all the vessels can communicate with a server. Moreover, an edge computing-based (EC) server is established to handle all the data for the vessels, including their locations and status values, in real time. So, the processing of the collected data on the EC server can help in making quick responses to abnormalities. The administrators on land communicate through the server, scheduling jobs, noticing abnormalities, and so on.

An experimental system was built on the existing VMS in in the East China Sea, which showed improved performance over current vessel monitoring systems. The average communication times was reduced and the real-time performance of the system improved. Moreover, the EC-VMS could improve the quality of data that is transmitted to shore.

The main contributions of this paper are as follows;

(1) Propose an edge computing-based framework of VMS, which can efficiently transmit the fishing vessels data and reduce the time of the network communication.
(2) A method based on Edge Computing is adopted to improve the real-time performance of the system in the case of marine restricted communication.
(3) Higher performance VMS compared to current systems.

2 Related Work

VMS can provide high resolution data on the spatial and temporal distribution of fishing effort. In Europe, the European Commission has introduced legislation to monitor fishing activity so that all vessels >15 m long are required to transmit their locations, estimated by GPS, at intervals of 2 h or less, so that the data is comparable with data provided by remote animal sensing [1].

The main drawback of VMS is that the data transmission is not in real-time. A large amount of sensor data can be generated on board, but cannot be fully transmitted to shore in time. So, VMS research is mostly focused on VMS data post-processing, to distinguish the employed fishing gear type [4], to detect potential fishing behavior from different gear types [5], to create fish abundance indices [6], to identify and characterize trips made by fishing vessels [7], and to improve fishing efficiency [8]. The other source of information was integrated to improving the uniformity of VMS Data, such as spaceborne high-resolution radar satellite data, satellite automatic identification system (sat-AIS) tracking data, and some vessel detection system (VDS) data [9].

Currently satellite communication is used in the maritime industry, however due to the limitations of satellite bandwidth, real time communications are affected and thus the performance of vessel monitoring systems degraded.

Recently, lots of progress has been made to improve the low-bandwidth communication in satellite positioning and satellite communication [10]. BDS was developed by China which can provide functions such as high precision positioning, short message communication, and Time services etc. In China, BDS is widely used in marine fishing vessels because of its low cost of short message communication [11]. Although there are many applications of marine communication system at present, there are still bottlenecks in the network, and the real-time performance is much worse than that on land [12–14].

Edge computing is becoming a new computing paradigm which combines edge IoT devices and cloud computing [15]. It processes data at the edge of the network, which has the potential to provide a better response time, battery life, bandwidth cost, data safety, and privacy. In edge computing, the computing occurs in the proximity of the data sources. Therefore, it has some advantages compared to cloud computing [16]. The results of some research have demonstrated these advantages [17–20]. The emerging edge computing technologies is the most important technique in our EC-VMS, which could achieve the goal of improving the response time and reduce the communication traffic.

The proposed framework of this paper has benefited from the edge computing paradigm to make the marine fishing management more real-time and intelligent.

3 Architecture

3.1 The Framework of EC-VMS

As shown in Fig. 1, the framework of EC-VMS mainly consists of four layers.

Perception Layer. There are many heterogeneous sensors, video surveillance, navigation and communication equipment in the ship. The perceptive layer refers to the

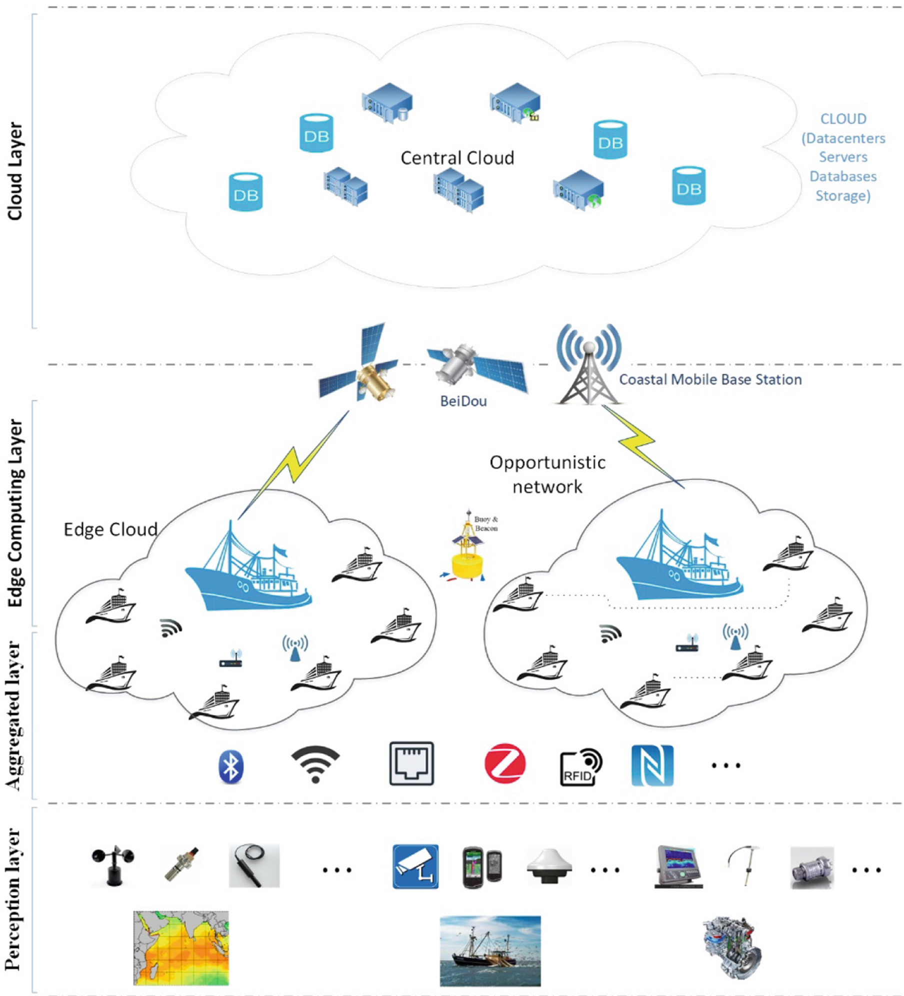

Fig. 1. The framework of EC-VMS

physical sensors and their running platforms. Through these devices, the perception layer gets the data of the operational state and the working environment of the ship.

Aggregated Layer. Shipborne data centers obtain the data for all the ship equipment through various application interfaces, preprocesses and stores them accordingly. The connection with the perceptive layer can be wired or wireless.

Edge Computing Layer. An edge computing-based management system is established between the ships, which can store and make decision immediately and in addition decides whether to forward information to cloud layer. The edge computing layer can run on only one ship, and it can also run in the form of ship network through a marine self-organizing mesh network.

Cloud Layer. A cloud computing-based management system is built on shore, which can store large amounts of ships data and manage the whole system. Moreover, the cloud layer can track all the ships in real time, make decisions and generate emergency commands.

3.2 Perception Layer

The Perception layer collects data mainly on three aspects of fishing vessels; marine environmental data, including meteorological, hydrological, sea surface temperature, humidity and salinity etc. Fishery production data including ship location, fishing conditions, fishing gear, fish catch, materials, personnel and video surveillance of operation etc. Equipment condition data including engine condition, oil quantity and the internal network etc.

Recently, RFID tags and various kinds of sensor technology are adopted by vessel builders. The RFID tag has a self-perception ability, which allows it to report its own status. The sensors can sample numerical values, which reflect the states of the monitored objects. Table 1 shows a part of the data that could be obtained from different sensors and devices onboard. These sampled numerical values reflect the states of the monitored objects.

Table 1. A part of the data from sensors and devices onboard.

Data	Category	Data type
Positioning & navigation	Fishery production	Numeric, characters, dates
Meteorological	Marine environmental	Numeric, image
Hydrological	Marine environmental	Numeric, image
Video surveillance	Fishery production	Video, audio
Power monitoring	Equipment condition	Numeric
Fishery administration	Fishery production	Numeric, characters, image

3.3 Aggregation Layer

The aggregation layer is an adaptor layer to connect the devices of perception layer, which is responsible for sensor node configuration, initialization, data acquisition, data caching and network manager. On modern vessels, the data sensors are shared over an Ethernet network available on the ship. All the local data obtained from sensors or devices onboard can be encapsulated and transmitted to the aggregated layer data storage center using different wireless protocols (e.g. WIFI, Bluetooth, ZigBee and UWB etc.). The aggregated layer has a data cache corresponding to the data cache of perception layer for each device, which is used to facilitate powerful distributed optimizations for communication.

There is an aggregated database to receive, store, and process the raw sampling data from the connected sensors, and then send the processed data to the edge computing layer. The database contains the basic data of vessels information, crew, navigation information, marine geographic information and fishery facilities etc. Moreover, it sets up the scheme for multi-source heterogeneous perception data (e.g. image data from video monitoring and trajectory data from GPS etc.). The aggregated layer exchanges and shares data with other vessels and provides data support for the edge computing layer.

3.4 Edge Computing Layer

The Edge Computing layer represents an abstract edge computer dedicated and responsible for a group of vessels. The Edge Computing layer and aggregation layer can overlap in their functions and both can co-exist within a network of vessels or on a single vessel. Data from the aggregation layer can be sent to the Edge Computing layer for storage, processing and analysis. In an edge computing environment, an aggregation layer can transmit data to its Edge Computing layer rapidly for analysis and respond to the perception device in a few seconds.

A larger aggregation Edge Computing layer that manages the services of local vessel networks is established in a selected vessel called Vessels Edge Computing Server (VECS), which can receive the data from a single aggregation layer in a vessel and make some advanced data analysis. In the larger aggregation edge computing network, vessels can also perform specific computations and communicate with each other. VECS decides which tasks go to the local edge computing node and which go to the cloud center.

In the EC-VMS, few sensor devices will transmit data directly to the cloud. The Edge Computing layer is mainly devoted to the vessel's local data processing and analyzing facilities for real-time needs such as emergency response services. Like the aggregation layer, the Edge Computing layer maintains both data and application caches which allow optimizations to be carried out by analyzing the interactions between sensor data and applications. Figure 2 shows the communication of EC-VMS.

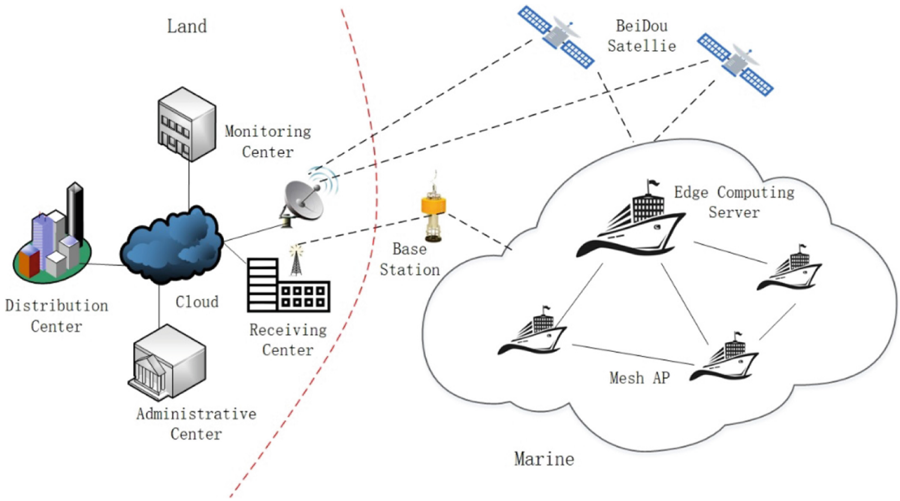

Fig. 2. The communication of EC-VMS

3.5 Cloud Layer

A Cloud layer is designed to provide central control, which delivers elastic computing power and storage at a low cost. However, cloud computing systems are shore based and therefore have an intrinsic delay due to processing and communication links. A local server allows for real time responses due the reduction in communications delay and its exclusive use for running the management system.

It is important to respond to the abnormal condition when the edge node becomes invalid. For example, if a vessel meets with a mishap, and the communication module is damaged, the ECS cannot receive the help message, but the cloud layer can give an alarm by running an anomaly detection service periodicity.

All the vessels in the EC-VMS are shown on the GIS for visualization. In addition, every vessel has its own information on the marine map, consisting of its name, unique ID, location, status and other attributes. Different colors are used to easily distinguish the different states. This makes it easy for administrative staff to see the abnormal vessels. Moreover, the situation must be display in real time. If one vessel is out of touch for a specified time, the vessel on the map must immediately be set to the color of the out of touch state. If a vessel is sailing into prohibited fishing areas, the vessel on the map should synchronously blink, and the message reported to relevant staff.

3.6 Interactivity Policy of EC-VMS

In this work vessel trajectory data was used to validate the EC-VMS, we use a transmission model called the Adaptable Trajectory Transmission Model (ATTM). ATTM combines the LDR algorithm [21], SQUISH trajectory compression algorithm [22] and reliable transmission strategy to establish a unified communication mechanism based on the EC-VMS. The model was divided into two parts in the edge computing layer; data tracking and data simplification.

In order to ensure that the trajectory can be transmitted to the ground monitoring center in time for real time analysis, the trajectory tracking and simplification must be synchronized. The ATTM uses synchronization mode so that when the tracking mechanism sends an updated trajectory, trajectory simplification will also be implemented.

Fishing vessels have a randomness in the process of operation, and its fishing behavior is complex. Therefore, the algorithms such as Neural Networks and Gauss Regression Processes are not suitable for track estimation. The LDR algorithm only needs base points and velocity vectors to estimate track.

This is a linear predictive function of the edge computing layer for the current position of fishing vessels.

$$\vec{l}(t) : t = l_b.\vec{p} + (t - l_b.t)\overrightarrow{l_V} \tag{1}$$

where l_b is the prediction base point, $\overrightarrow{l_V}$ is velocity vector. For a given error threshold θ_d, LDR guarantees that when the predicted trajectory point P'_t are close to the observation trajectory point P_t, that is $\mathrm{ED}\left(P_t, P'_t\right) < \theta_d$, the edge computing layer will not produce update messages, and the shore-based monitoring center uses the predicted points instead of the observation points. If the observed trajectory deviates from the predicted trajectory then the prediction base point and velocity vector need to be updated.

In the case of frequent trajectory changes, the edge computing layer needs to send more trajectory points. However, the BDS communication protocol has strict restrictions on message length and transmission time interval, so we need to select a fixed-length trajectory sequence (adapted to the BDS protocol packet) T' from the original observation trajectory T, and send it to cloud layer. ATTM uses SQUISH algorithm for selection, because SQUISH runs fast, has good real-time performance, and can preset the size of the approximate trajectory sequence. The edge computing layer adds the observed trajectory points to the buffer of the SQUISH algorithm. When the transmission condition is reached, the fixed size trajectory sequence T' is obtained from the buffer and sent to the cloud layer together with the update message.

Algorithm 1: ATTM (edge computing layer)

Input:
(1) error threshold θ_d
(2) observation trajectory point P_t
Function:
send messages
Begin

```
initial uncompressed queue;
initial sending queue;
while (received data)
    if received a retransmit signal then
        adding missing messages to the sending queue
        based on message number;
    if received the observation trajectory points then
        if the uncompressed queue is empty then
            estimate trajectory points by LDR;
            if estimated value greater than threshold then
                add observation point to uncompressed queue;
        else add observation point to compressed queue;
    if it's time window for data transmission then
        if sending queue is not empty then
            send message;
        else if uncompressed queue is not empty then
            compress trajectory by SQUISH;
            generate message into sending queue;
            send message;
```

End

The cloud layer uses the same trajectory estimation algorithm as the edge computing layer to display the ship's position in real time. In order to reduce the number of satellite communications, the cloud layer will not send a communication receipt for each received message. The Cloud layer updates existing trajectory data according to the new messages.

4 Result and Analysis

4.1 Experimental Setup

The experimental data was collected from the trajectory data of four fishing vessels in the VMS that took place in the East China Sea, near Zhoushan City, Zhejiang Province, China. The VMS manages more than 3,000 vessels. This trajectory data is generated by the shipborne BDS terminal module, and the device can collect positional data once a second, but the minimum interval of satellite transmission is limited to 60 s. Trajectory data contains information such as device number, time, longitude and latitude. In order to control the experimental variables and improve the accuracy of the experiment, we chose four vessels and installed edge computing nodes. The edge computing nodes collected complete trajectory data of four fishing vessels from March 2018 to May 2018, totaling 1018412 trajectories' points. The spatial distribution of the four vessels are shown in Fig. 3.

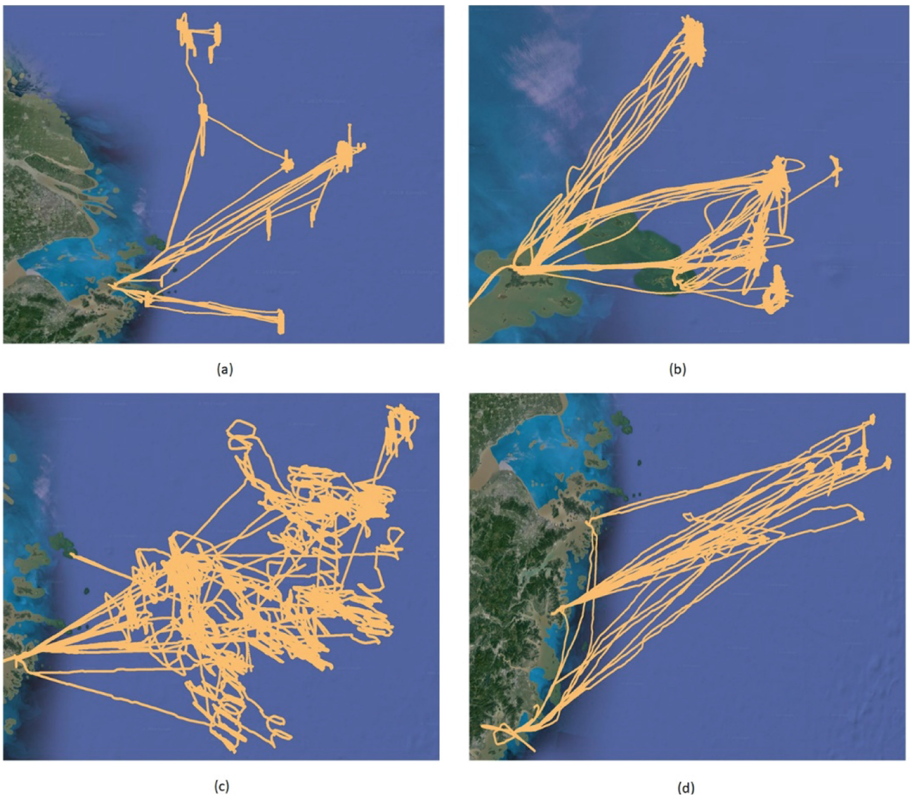

Fig. 3. Spatial distribution of four vessels' trajectory.

This paper uses the ATTM algorithm to verify the framework proposed in this work, which considers the limitation of the BDS communication protocol on message length and minimum transmission interval. When the transmission interval does not reach the minimum transmission interval, it is not allowed to send messages. When the message length exceeds the maximum transmission length, the data beyond the maximum transmission length will be discarded. Meanwhile, this paper also considers the situation of message distortion and packet loss.

4.2 Experimental Results

The experiment is analyzed from three aspects; the number of trajectory data transmission, the real-time performance and the trajectory quality. Figure 4 shows the comparison of ATTM transmission times with the traditional fixed-interval transmission mode (FITM) of VMS in three cases: 30-m threshold, 50-m threshold and 70-m threshold. FITM transferred data at each time interval. The abscissa represents the minimum communication interval of the VMS, and the ordinate represents the number of communications. As can be seen from the figure, the communication times of FITM and ATTM decrease with an increase in the communication interval.

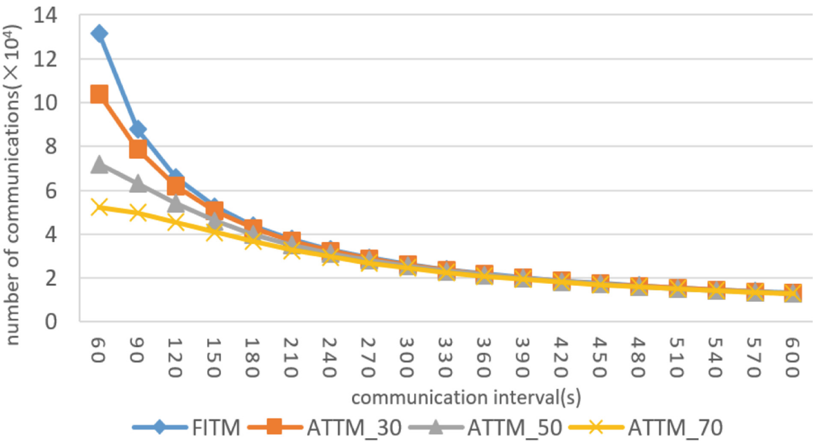

Fig. 4. Comparison of transmission times.

ATTM has a low probability of predicting all observation trajectory points accurately when the communication interval is large. It needs to communicate every time when it reaches the communication window, so the number of transmissions decreases slightly, which is close to FITM. Meanwhile the criterion of accurate prediction is that the distance between the observation trajectory point and the prediction trajectory point is less than the threshold, so the larger the threshold, the less the number of communications. ATTM has less communication times than FITM protocol under different communication intervals and error thresholds, so it has obvious effect in saving communication resources. Under the typical 60-s transmission interval and 50-m threshold, the network traffic is reduced by 45.22%.

Real-time trajectory query is another important indicator of EC-VMS. FITM transmits data at fixed time intervals. When the cloud layer receives data at time t1, it needs to wait for data at time t1 + 1. Therefore, the minimum delay time of FITM query is 0 s, the maximum delay time is the transmission time interval Δt, and the average delay time is Δt/2 s. The ATTM protocol can be used for real time analysis, however there is an intrinsic delay in the system as the trajectory data will only be updated when the cloud service receives the updated data. In order to compare with FITM, this paper uses statistics to analyze the trajectory data correction time.

As can be seen from Fig. 5, the correction time of ATTM increases with the communication time interval. This is because when the communication interval is large, the ATTM cannot send the correction information in time, which leads to a higher delay time. The higher the error threshold is, the fewer trajectory points are needed to be corrected, so the real-time performance is better. The communication interval and error threshold directly affect the real-time performance of ATTM. It can be seen from Fig. 5 that the correction time of ATTM is obviously lower than FITM, so we can conclude that the real-time performance of ATTM is better than FITM.

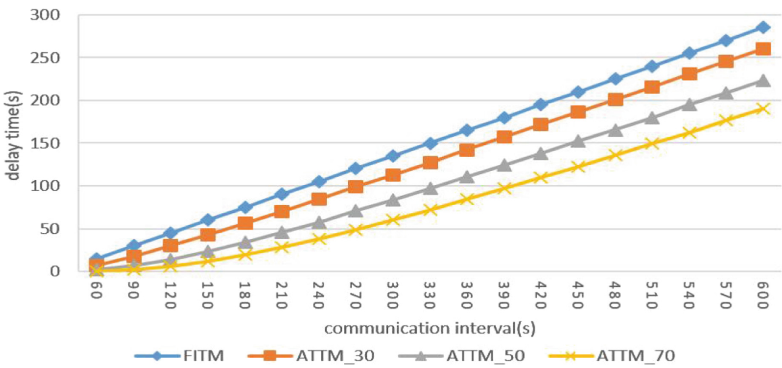

Fig. 5. Comparison of real-time performance.

In order to compare the trajectory data quality of ATTM and FITM, we use the Average of Pairs Distance (APD) as the evaluation criterion. Given trajectories A and B, APD calculates the distance between the points corresponding to the two trajectories and calculates the average value. The calculation formula is as follows:

$$\mathrm{APD}(\mathrm{A},\mathrm{B}) = \frac{1}{n} \times \sum\nolimits_{i=1}^{n} ED(a_1, b_1) \quad (2)$$

In this experiment, A is the trajectory queried in VMS and B is the original observation trajectory. The results are shown in Fig. 6.

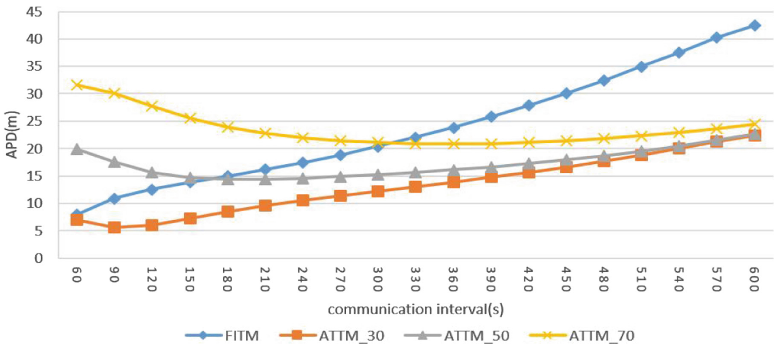

Fig. 6. Comparison of trajectory quality.

We set the error threshold of ARTT to 30 m, 50 m and 70 m, and compared it with FITM.

The APD of FITM increases as the communication interval gets larger due to the lower number of trajectory points in the FIFM transmission. In Fig. 6, the ADP of ARTT decreases first and then gradually increases. This is the result of a large number of points which have been calculated incorrectly being transmitted when the communication interval is small. This means the LDR algorithm is used more frequently and SQUISH compression algorithm is used less frequently, which makes the prediction error larger than the compression error.

With the increase of the communication interval, the proportion of prediction points decreases and the APD decreases. As the interval continues to increase, the proportion of compressed points increases, and the error caused by compression also increases, which eventually leads to an increasing trend of ADP. The larger the error threshold of ARTT is, the larger the value of the ADP will be. In the case of a 30 m error threshold, ATTM has a significant improvement over the FITM trajectory quality.

5 Conclusion

In order to reduce the communication cost and improve real-time efficiency of the VMS, we propose a framework of edge computing-based VMS in this paper. In the EC-VMS, firstly, in order to get more data, a perception platform is established on every vessel to interactive with the data collector. Therefore, it can monitor itself in real-time and provide the data support for the server. Secondly, the EC-VMS adopts the BDS for communication because of its low price and wide coverage. Thus, all the vessels can communicate with the server. Thirdly, an edge computing-based server is established to handle all the data for the vessels, including their locations and status values, in real time. So, the processing of the collected data on the edge computing server can help in making a quick response. Moreover, a data transmission model called ATTM was established to interact between the cloud and edge. The experiment is based on the data of an existing VMS that runs in the East China Sea, Zhoushan City. Results show that it is better than the original VMS in real-time, efficiency and usability. In the future work, more types of vessels data and edge computing methods will be investigated.

Acknowledgment. This work was supported in part by the Key Research and Development Project of Zhejiang Province (Grant No. 2017C03024), the National Natural Science Foundation of China (Grant No. 61572163) and the Zhejiang Province Research Program (Grant No. 2017C3 3065).

References

1. Stephen, C.V., Stuart, B., Matthew, J.W., Richard, I., David, T., Jason, N.: Individual responses of seabirds to commercial fisheries revealed using GPS tracking, stable isotopes and vessel monitoring systems. J. Appl. Ecol. **47**(2), 487–497 (2010)
2. Lee, J., South, A.B., Jennings, S.: Developing reliable, repeatable, and accessible methods to provide high-resolution estimates of fishing-effort distributions from vessel monitoring system (VMS) data. ICES J. Mar. Sci. **67**(6), 1260–1271 (2010)
3. Ejaz, A., Mubashir, H.R.: Mobile edge computing opportunities, solutions, and challenges. Future Gener. Comput. Syst. **70**, 59–63 (2017)
4. Marzuki, M.I., Gaspar, P., Garello, R.: Fishing gear identification from vessel-monitoring-system-based fishing vessel trajectories. IEEE J. Oceanic Eng. **43**(3), 689–699 (2018)
5. de Souza, E.N., Boerder, K., Matwin, S.: Improving fishing pattern detection from satellite AIS using data mining and machine learning. PLOS ONE **11**(7), e0158248 (2016)
6. Ducharme-Barth, N.D., Shertzer, K.W., Ahrens, R.N.M.: Indices of abundance in the Gulf of Mexico reef fish complex: a comparative approach using spatial data from vessel monitoring systems. Fish. Res. **198**, 1–13 (2018)
7. Watson, J.T., Haynie, A.C.: Using vessel monitoring system data to identify and characterize trips made by fishing vessels in the United States North Pacific. PLOS ONE **11**(10), e0165173 (2016)
8. Watson, J.T., Haynie, A.C., Sullivan, P.J.: Vessel monitoring systems (VMS) reveal an increase in fishing efficiency following regulatory changes in a demersal longline fishery. Fish. Res. **207**, 85–94 (2018)
9. Longepe, N., Hajduch, G., Ardianto, R.: Completing fishing monitoring with spaceborne Vessel Detection System (VDS) and Automatic Identification System (AIS) to assess illegal fishing in Indonesia. Marine Pollution Bulletin **131**(SI), 33–39 (2018)
10. Al-Zaidi, R., Woods, J., Al-Khalidi, M.: Next generation marine data networks in an IoT environment. In: Second International Conference on Fog and Mobile Edge Computing 2017, FMEC, pp. 50–55. IEEE, Valencia (2017)
11. Lu, C., Li, X., Nilsson, T.: Real-time retrieval of precipitable water vapor from GPS and BeiDou observations. J. Geodesy **89**(9), 843–856 (2015)
12. Zhang, Y., Chen, S., Hong, Z.: Feasibility of oil slick detection using BeiDou-R coastal simulation. Math. Prob. Eng. **4**, 1–8 (2017)
13. Yu, F., Hu, X., Dong, S.: Design of a low-cost oil spill tracking buoy. J. Mar. Sci. Technol. **23**(1), 188–200 (2018)
14. Wang, L., Li, L., Qiu, R.: Edge computing-based differential positioning method for BeiDou navigation satellite system. KSII Trans. Internet Inf. Syst. **13**(1), 69–85 (2019)
15. Satyanarayanan, M.: The emergence of edge computing. Computer **50**(1), 30–39 (2017)

16. Shi, W., Cao, J., Zhang, Q.: Edge computing: vision and challenges. IEEE Internet Things J. **3**(5), 637–646 (2016)
17. Zeydan, E., Bastug, E., Bennis, M.: Big data caching for networking: moving from cloud to edge. IEEE Commun. Mag. **54**(9), 36–42 (2016)
18. Rahmani, A.M., Gia, T.N., Negash, B.: Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: a fog computing approach. Future Gener. Comput. Syst. **78**, 641–658 (2018)
19. Taleb, T., Dutta, S., Ksentini, A.: Mobile edge computing potential in making cities smarter. IEEE Commun. Mag. **55**(3), 38–43 (2017)
20. Premsankar, G., Di Francesco, M., Taleb, T.: Edge computing for the internet of things: a case study. IEEE Internet Things J. **5**(2), 1275–1284 (2018)
21. Trajcevski, G., Cao, H., Scheuermann, P., Wolfson, O., Vaccaro, D.: On-line data reduction and the quality of history in moving objects databases. In: Proceedings of the 5th ACM International Workshop on Data Engineering for Wireless and Mobile Access, MobiDE 2006, pp. 19–26. ACM, Chicago (2006)
22. Muckell, J., Hwang, J., Patil, V., Lawson, C., Ping, F., Ravi, S.: SQUISH: an online approach for GPS trajectory compression. In: Proceedings of the 2nd International Conference and Exhibition on Computing for Geospatial Research & Application, COM. Geo 2011. ACM, Washington DC (2011)