

## Research Article

# Distributed Measurement Data Gathering about Moving Objects

Ivan Kholod, Nikolay Plokhoy, and Andrey Shorov

*Faculty of Computer Science and Technology, Saint Petersburg Electrotechnical University "LETI", Professora Popova Str. 5, Saint Petersburg, Russia*

Correspondence should be addressed to Ivan Kholod; [iikholod@mail.ru](mailto:iikholod@mail.ru)

Received 28 July 2017; Revised 3 November 2017; Accepted 23 November 2017; Published 19 December 2017

Academic Editor: Santiago Mazuelas

Copyright © 2017 Ivan Kholod et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper describes approaches to gathering measurement data about moving objects in networks with low bandwidth. The first approach uses Fog computing conception and suggests moving assessing the quality of the measurement data into measuring points. The second approach uses prediction of telemetry quality by mining models. In addition, the paper presents implementation of these approaches based on actor model. As a result, it became possible not only to load balancing among edge and cloud nodes, but also to significantly reduce the network traffic, which in turn brings the possibility of decreasing the requirements for communication channels bandwidth and of using wireless networks for gathering measurement data about moving objects.

## 1. Introduction

The task of measurement data gathering from moving objects (MOs) is a very relevant one. The following examples are given to illustrate this:

- (i) In the transportation industry, telemetry provides meaningful information about driver's performance by collecting data from a vehicle, leading to higher fuel consumption efficiency through driver's feedback, which includes in-cab coaching.
- (ii) Space science measurements are used by manned or unmanned spacecraft for data transmission.
- (iii) Rocketry measurement equipment forms an integral part of the rocket range assets used to monitor the position and health of a vehicle launch.
- (iv) Today nearly every type of aircraft, missiles, or spacecraft carries a wireless telemetry system as it is tested. Aeronautical mobile telemetry is used for the safety of pilots and persons on the ground during flight tests. Telemetry from an on-board flight test instrumentation system is the primary source of real-time measurement and status information transmitted during the testing of manned and unmanned aircraft.
- (v) Motor racing measurement is a key factor in modern motor racing, allowing race engineers to interpret data collected during a test or a race and use it to properly tune a car for optimum performance.

This task has become even more relevant with development of the unmanned vehicles industry. Measurements (speed, fuel availability, temperature, etc.) transferred from the vehicle to the control point using telemetry channels are used for unmanned vehicle management.

Measurement data gathering about MO involves the use of measuring points (MPs) located along the route of the movement of the MO. Each of them measures various parameters of the objects' performance. Among them we can distinguish the following kinematic parameters: coordinates  $(x, y, z)$  and the velocity vector  $(vx, vy, vz)$ ; parameters transferred through the telemetry channel: temperature, vibration, fuel level, and so on. Measurements can be obtained by different measuring systems (MSs): radar, lidar, optics, telemetry, navigation, and so on.

All measurement data that have been received from MSs are continuously transferred to the measurements processing center (MPC) (Figure 1). At the MPC, the measurement data are combined into a main stream of measurements. This stream can be used to identify potential hazardous situations with MO and make decisions to manage them. In addition, tuning of MS can be performed based on the results of

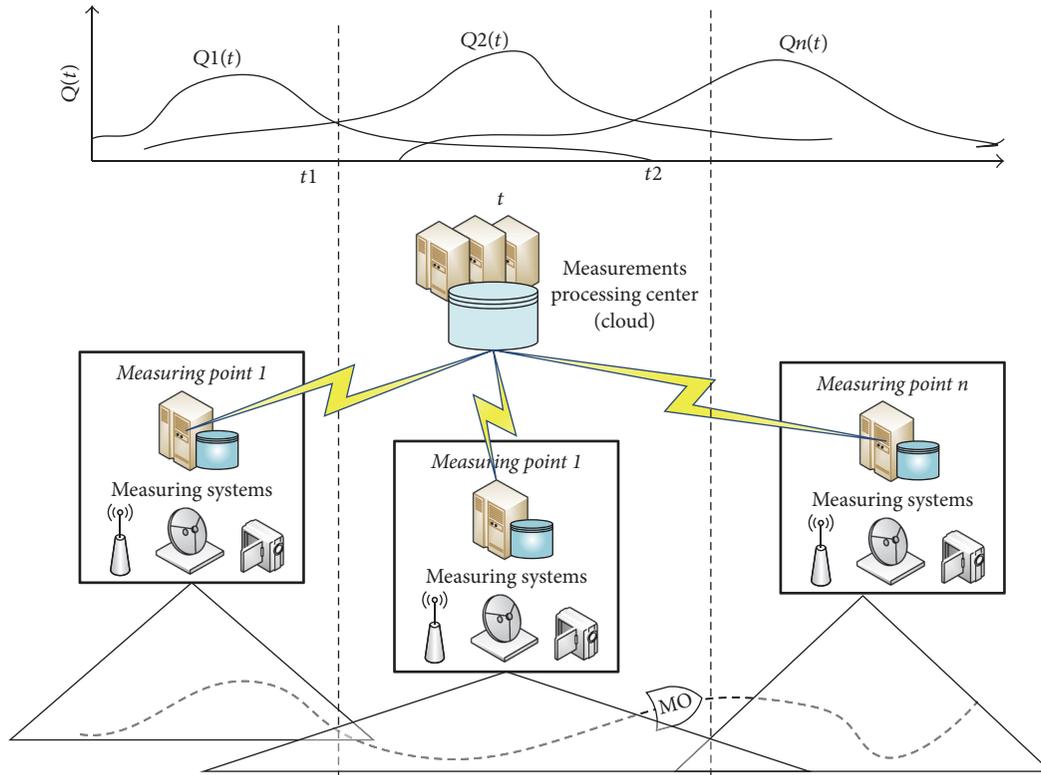


FIGURE 1: Gathering of measurement data from moving objects.

measurement processing (e.g., aiming of radar antennas and camera lenses) using current and predicted parameters of MO (its location, speed, etc.). A MPC can be created based on cloud computing technology.

Typically, during measurements, all MSs work in the measurement mode, regardless of the presence of the object in their detection zones. It is necessary that measurements of an object are taken as soon as it enters the MS detection zone. However the detection zones are different for each MP and they may have significant intersections (Figure 1). As a consequence, the MPC receives duplicate data of different quality ( $Q$ ) from all MPs throughout the period of measurement. Thus, the collected data have a significant percent of duplicated measurements and noise.

Thus the following problems of measurement data gathering and processing arise during measurement:

- (i) Geographical isolation of MPs from each other and from the MPC
- (ii) Duplication of data received from different MPs, as their detection zones intersect
- (iii) Need to control both MSs and MO itself, requiring real-time analysis of the collected data
- (iv) Using wireless networks with limited bandwidth (satellite, radio relay, cellular, etc.).

More recently, wireless networks, including cell phone networks, wireless local area networks (WLANs), wireless sensor networks, satellite communication networks, and

terrestrial microwave networks, have increasingly been used for data gathering, including telemetry [1]. This is explained by convenience of their use: no need for cable installation, ease of connection, and so on. However, one of the principal disadvantages of networks of this type is limited bandwidth.

This paper proposes to use dynamic management of measurement data streams in order to decrease network traffic by transferring only necessary measurements. It allows reducing the volume of transferred data and decreasing bandwidth requirements for communication channels.

## 2. Related Work

The problem of remote data (telemetry) gathering is not new. First data-transmission circuits go live between the Russian Tsar's Winter Palace and army headquarters in 1845. At the end of the last century and at the beginning of this century, the concept of Machine to Machine communications (M2M) became widespread. M2M technology continues where telemetry left off: wireless telemetry systems today cost substantially less compared to what they did 10 to 15 years ago. Applications that were previously not economically feasible are now cost effective. M2M brings discipline to telemetry through open standards and protocols.

At the present time, we can observe rapid growth in the number of the Internet of Things (IoT) devices. The basic idea of IoT is to connect all things (devices) in the world to the Internet. According to Gartner, Inc. (a technology research and advisory corporation), there will be nearly 26 billion devices on the Internet of Things by 2020 [2].

The basic architecture of IoT which is widely used to explain the approaches of IoT has three layers [3]:

- (i) The perception layer is the bottom layer which can be regarded as the hardware or physical layer which does the data collection.
- (ii) The network layer (the middle layer) is responsible for connecting the perception layer and the application layer so that data can be passed between them.
- (iii) The application layer usually plays the role of providing services or applications that integrate or analyze the data received from the other two layers.

Some researchers [4] propose to extend the layers. They add new layers:

- (i) Access gateway layer takes care of message routing, publishing, and subscribing and also performs cross platform communication, if required.
- (ii) Middleware layer acts as an interface between the hardware layer at the bottom and the application layer at the top and is responsible for critical functions such as device management and information management and also takes care of issues like data filtering, data aggregation, semantic analysis, access control, and information discovery.

The data processing and analysis services for IoT are deployed on the top layers and divide them into the following sublayers [5, 6]:

- (i) Data gathering layer manages collected data: object identification, data abstraction, compression, parsing, and merging; various data are saved in the corresponding database or data warehouse.
- (ii) Data processing layer executes data preprocessing and filtering of observed events and allows us to aggregate, organize, and analyze data according to events.

Usually, for implementation of the application and middleware layers, cloud computing technologies are used [6]. A cloud provides scalable storage, computation time, and other tools to build application services. It can be also used to process telemetry gathering from objects.

Similar architecture is also applied for measurement data gathering about moving objects. For example, space situational awareness (SSA) program in European space agency (ESA) and the Egyptian space program use cloud computing for telemetry processing [7]. It enables software and hardware decoupling and makes flexible telemetry data analysis possible. The large amount of available computational resources facilitates a shift in approaches to software development, deployment, and operations. This approach was also used in the Cloud-Based Ground System for space telemetry processing [8].

In this case, the network layer is responsible for connecting the IoT devices (such as sensors, radar, cameras, and other measuring systems) and a cloud. It creates very large network traffic. Solution of this problem can be Fog computing [9].

The fog extends the cloud to be closer to the sources that produce and act on IoT data.

Fog computing either completely solves or decreases the influence of common problems in distributed systems:

- (i) Big delays in the network
- (ii) Scaling data sources
- (iii) Problems related to endpoints mobility
- (iv) High cost of the broadband
- (v) Wide geographical spread of the systems.

Despite Fog computing becoming popular, there are no ready solutions for its implementation. This can be explained by the fact that such concept is very young and has a high level of abstraction. The paper describes approaches to measurements data gathering based on Fog computing technology. These approaches suggest moving part of computations closer to MPs. It allows redistributing the computational load and reducing network traffic that it is very important to use wireless networks for measurements data gathering.

### 3. Generic Approach to Telemetry Gathering

*3.1. Formal Representation of a Measurement Data Gathering System from Moving Object.* The entire measurements data gathering system can be represented formally as a set:

$$GS = \langle mo, mpc, MP \rangle, \quad (1)$$

where

- (i) mo is a controlled moving object;
- (ii) mpc is a MPC; MP is a set of MPs.

The state of any MO is characterized by a set of parameters:

$$\begin{aligned} S_{mo} &= S_{mo}^k \cup S_{mo}^t \\ &= \{x, y, z, vx, vy, vz\} \cup \{p_1, p_2, \dots, p_k, \dots, p_u\}, \end{aligned} \quad (2)$$

where

- (i)  $S_{mo}^k$  are kinematic parameters: coordinates  $(x, y, z)$  and the velocity vector  $(vx, vy, vz)$ ;
- (ii)  $S_{mo}^t$  are parameters  $p_k$ ,  $k = 1 \dots u$ , transferred through the telemetry channel: temperature, vibration, fuel level, etc.

The MP is a set of MPs:

$$MP = \{mp_1, mp_2, \dots, mp_r, \dots, mp_m\}. \quad (3)$$

Each MP  $mp_r$  includes MSs that measure kinematic parameters of MO and MSs receiving telemetry data from MO. Therefore, a vector of MO parameters is formed at each measuring point  $mp_r$  at each time instant  $t_j$ :

$$\begin{aligned} s_r(t) &= s_r^k(t) \cup s_r^t(t) \\ &= \{x_r(t), y_r(t), z_r(t), vx_r(t), vy_r(t), vz_r(t)\} \\ &\quad \cup \{p_{r.1}(t), \dots, p_{r.k}(t), \dots, p_{r.u}(t)\}, \end{aligned} \quad (4)$$

where

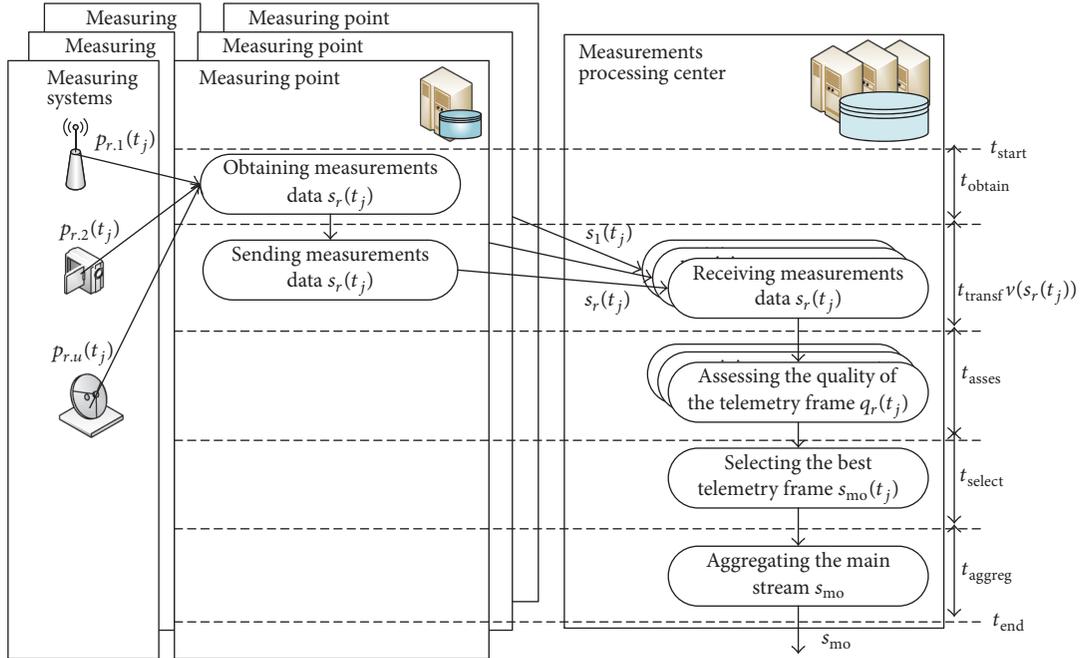


FIGURE 2: Diagram of the approach with full measurement data gathering (FG).

- (i)  $s_r^k(t_j)$  is a vector of kinematic parameters obtained at time  $t_j$  at the point  $mp_r$ ;
- (ii)  $s_r^t(t_j)$  is a telemetry frame containing a set of parameters of the object MO obtained at time  $t_j$  by telemetry at the point  $mp_r$ .

For each telemetry frame  $s_r^t(t_j)$ , it is possible to measure its quality  $q_r(t_j)$ , characterizing the integrity and reliability of the received data. The quality of telemetry frame can be estimated in many ways and is not considered in this paper (verification of built-in markers (e.g., Reed-Solomon codes) for each frame can serve as an example of estimation of telemetry data quality).

The main purpose of a measuring system is to produce a main stream containing the best measurements of all parameters of a MO at each time instant of the measurement:

$$s_{mo} = \{s_{mo}(t_1), s_{mo}(t_2), \dots, s_{mo}(t_j), \dots, s_{mo}(t_w)\}, \quad (5)$$

where  $s_{mo}(t_j)$  is the set of the best values of MO parameters at time  $t_j$ .

Thus, the generic approach with full measurement data gathering (FG) involves the following procedure performed at each time instant  $t_j$  (Figure 2):

- (1) Obtaining measurements data  $s_r(t_j)$  on each MP  $mp_r$ ,  $r = 1 \dots m$
- (2) Transferring measurements data  $s_r(t_j)$  from each MP  $mp_r$ ,  $r = 1 \dots m$  into the MPC
- (3) Assessing the quality of the telemetry frame  $q_r(t_j)$  received from each MP
- (4) Selecting the best telemetry frame  $s_{mo}(t_j)$
- (5) Aggregating the main stream  $s_{mo}$ .

The runtime of a data transfer operation is determined (based on Hockney's model [10]) by the volume of the data  $v$ , the latency  $\alpha$ , and the bandwidth  $\beta$  of a communication channel:

$$t_{transf}(d) = \alpha + \frac{v(d)}{\beta}, \quad (6)$$

where  $v(d)$  is volume of data  $d$ .

The total time for the formation of the main stream  $T_{FG}$  (from  $t_{start}$  to  $t_{end}$ ) at each instant time  $t_j$  can be calculated using the following formula:

$$T_{FG} = \max_{r=1}^m(t_{obtain}) + \max_{r=1}^m(t_{transf}(s_r(t_j))) + \sum_{r=1}^m t_{asses} + t_{select} + t_{aggregate}. \quad (7)$$

Network traffic for a data-transmission operation is determined by the amount of transferred data. Therefore, network traffic  $V_{FG}$  at each instant time  $t_j$  is determined by the following formula:

$$V_{FG}(t_j) = \sum_{r=1}^m v(s_r(t_j)). \quad (8)$$

The quality of the best telemetry frame  $Q(t_j)$  at each instant time  $t_j$  is determined by the maximum quality among frames gathering from all MPs:

$$Q_{FG}(t_j) = \max_{r=1}^m(q_r(t_j)). \quad (9)$$

The advantage of the FG approach is the highest possible quality of telemetry because it gathers telemetry from all MPs and selects the best frames from them. The disadvantages of this approach are as follows:

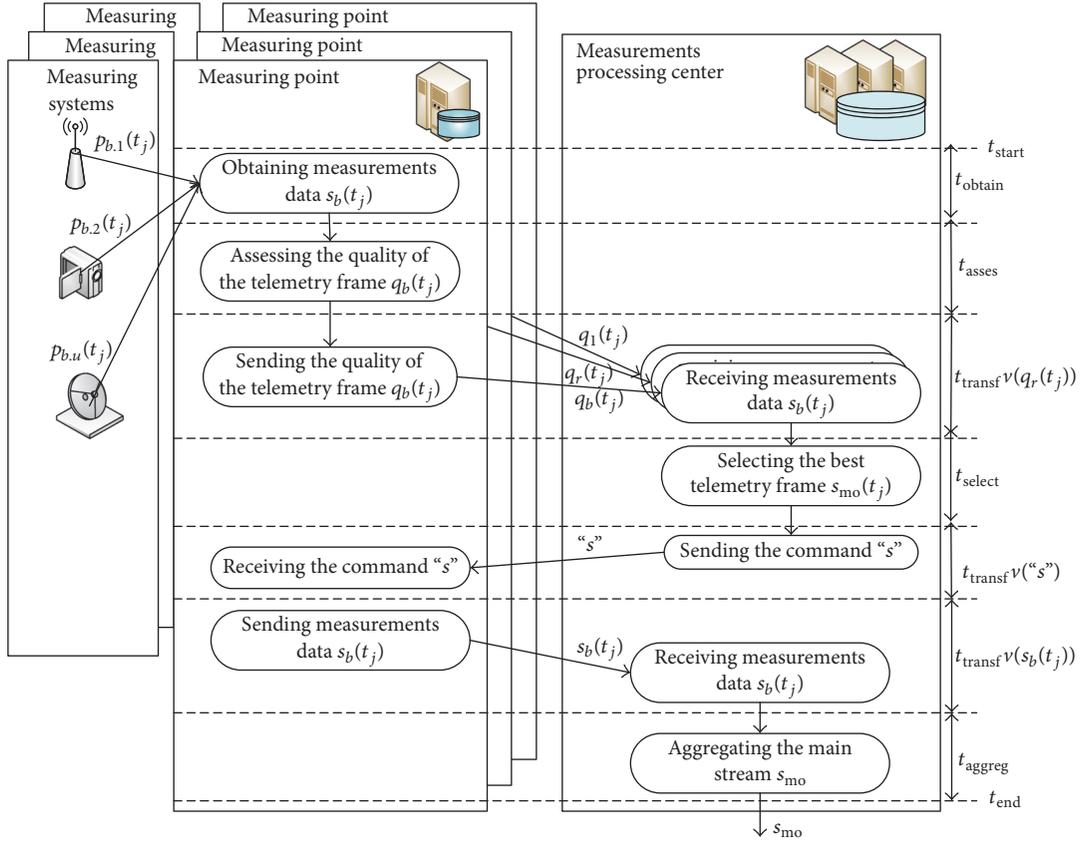


FIGURE 3: Diagram of the approach with a distributed quality assessment (DQA).

- (i) High network traffic  $V_{FG}$  between MP and the MPC due to the need to transfer all measurements from MP, even those that will not eventually be included in the main measurement stream
- (ii) A large number of calculations in the MPC associated with the quality evaluation of the incoming streams from each  $mp_r \in MP$ , selection of the best stream, and their aggregation into the main stream.

The situation becomes even more difficult if there are several MOs and measurement streams multiply depending on the number of MOs.

**3.2. Gathering with a Distributed Quality Assessment.** In order to reduce network traffic during measurements data gathering, it is proposed to transfer only the telemetry frame that will eventually be included into the main stream. To achieve this, the telemetry frame quality must be assessed at the MP and transferred to the MPC. The MPC collects assessments of quality and selects the best one. The best telemetry frame is requested from the corresponding MP and sent to the MPC to be included into the main stream.

Thus, the following procedure is performed at each time instant  $t_j$  (Figure 3):

- (1) Obtaining measurements data  $s_r(t_j)$  on each MP  $mp_r$ ,  $r = 1 \dots m$

- (2) Assessing the quality of the telemetry frame  $q_r(t_j)$  on each MP  $mp_r$ ,  $r = 1 \dots m$
- (3) Transferring the quality of the telemetry frame  $q_r(t_j)$  from each MP  $mp_r$ ,  $r = 1 \dots m$  into the MPC
- (4) Selecting the best telemetry frame
- (5) Transferring the command "s" to the selected MP  $mp_b$  to transfer the telemetry frame
- (6) Transferring the best telemetry frame  $s_b(t_j)$  from MP  $mp_b$  to the MPC
- (7) Aggregating the main stream  $s_{mo}$ .

The total time for the formation of the main stream  $T_{DQA}$  at each instant time  $t_j$  can be calculated using the following formula:

$$\begin{aligned}
 T_{DQA} = & \max_{r=1}^m (t_{obtain}) + \max_{r=1}^m (t_{asses}) \\
 & + \max_{r=1}^m (t_{transf}^v(q_r(t_j))) + t_{select} \\
 & + t_{transf}^v("s") + t_{transf}^v(s_b(t_j)) \\
 & + t_{aggregate}.
 \end{aligned} \tag{10}$$

Network traffic for data-transmission operations is determined by the amount of data transferred. Therefore, network

traffic  $V_{\text{DQA}}$  at each instant time  $t_j$  is determined by the following formula:

$$V_{\text{DQA}}(t_j) = \sum_{r=1}^m v(q_r(t_j)) + v("s") + v(s_r(t_j)). \quad (11)$$

The quality of the best telemetry frame  $Q(t_j)$  at each instant time  $t_j$  is also determined by the maximum quality of the snapshots collected from all measuring points:

$$Q_{\text{DQA}}(t_j) = \max_{r=1}^m (q(s_r^t(t_j))). \quad (12)$$

This approach makes it possible to decrease network traffic significantly, because

$$\sum_{r=1}^m v(q_r(t_j)) + v("s") + v(s_r(t_j)) \ll \sum_{r=1}^m v(s_r(t_j)). \quad (13)$$

In addition, calculations required to assess the quality of telemetry frame are performed on each MP, which significantly decreased the load at the MPC.

The disadvantage of approach with a distributed quality assessment (DQA) is a possible increase in the main stream formation time after the moment of measurement, because additional functions are executed:

$$\begin{aligned} & \max_{r=1}^m (t_{\text{transf}}(s_r(t_j))) + \sum_{r=1}^m t_{\text{asses}} \\ & < \max_{r=1}^m (t_{\text{asses}}) + \max_{r=1}^m (t_{\text{transf}}(v(q_r^t(t_j)))) \\ & + t_{\text{transf}}(v("s")) + t_{\text{transf}}(v(s_b(t_j))). \end{aligned} \quad (14)$$

**3.3. Gathering with Quality Assessment Prediction.** In order to reduce the main stream formation time, it is suggested to predict quality of telemetry frame at each time instant  $q(s_r^t(t_j))$ . By predicting the change in the quality of the telemetry, it is possible to preemptively determine the time instants when it is necessary to transfer a frame to be included in the main stream.

The main problem is to calculate  $q(s_r^t(t_{j+n}))$  for a period of time  $n$  sufficient for MS switching. Such prediction can be achieved by applying mining models, obtained through the use of data mining algorithms, including those used for time series analysis (logical regression, etc.).

In this case, quality value  $q(t_{j+n})$  will be approximated for the following period. However, to improve the accuracy of the prediction, not only the existing values, but also additional attributes that affect the quality of the telemetry can be used. In this case, it is possible to produce the following vector that would characterize the quality of telemetry frame  $s_r^t(t_j)$  at time instant  $t_j$  from MP  $\text{mp}_r$ :

$$\begin{aligned} x_r(t_j) = \{ & x(t_j), y(t_j), z(t_j), vx(t_j), vy(t_j), vz(t_j), \\ & q(s_r^t(t_j)), q(s_r^t(t_{j+n})) \}. \end{aligned} \quad (15)$$

Thus training set is the time-ordered sequence of all vectors from the start time instant of measurement  $t_0$  till the finish  $t_w$  on each MP  $\text{mp}_r$ :

$$\begin{aligned} X_r = \{ & x_r(t_0), x_r(t_1), \dots, x_r(t_j), \dots, x_r(t_{j+n}), \dots, \\ & x_r(t_w) \}. \end{aligned} \quad (16)$$

A mining model can be constructed by data mining algorithms using such vector sets obtained from previous measurements of this MP's objects. In this case, the quality assessment  $q(s_r^t(t_{j+n}))$  is known for each  $x_r(t_j)$ . In addition, the mining model can be corrected in the course of measurements, using the measurements that have already been performed.

Using approach with a quality assessment prediction (QAP), the formation of the main stream  $s_{\text{mo}}(t)$  at each time instant  $t_j$  is performed concurrently on each MP  $\text{mp}_r$  and at the MPC (Figure 4):

(i) Previously, the following actions are performed on each MP  $\text{mp}_r$ ,  $r = 1 \dots m$ :

- (1) Obtaining measurements data  $s_r(t_j)$  on each MP  $\text{mp}_r$ ,  $r = 1 \dots m$
- (2) Assessing the quality of the telemetry frame  $q_r(t_{j-1})$  on each MP  $\text{mp}_r$ ,  $r = 1 \dots m$
- (3) Sending the quality assessment  $q_r(t_{j-1})$  and kinematic parameters  $s_r^k(t_{j-1})$  from each MP  $\text{mp}_r$ ,  $r = 1 \dots m$  into the MPC.

(ii) Previously, the following actions are performed at MPC:

- (1) Receiving the quality assessment  $q_r(t_{j-1})$  and kinematic parameters  $s_r^k(t_{j-1})$  from each MP  $\text{mp}_r$ ,  $r = 1 \dots m$
- (2) Predicting the quality of the telemetry frame  $q'_r(t_j)$  for each MP  $\text{mp}_r$ ,  $r = 1 \dots m$  at time  $t_j$
- (3) Selecting the best MP  $\text{mp}_b$  at time  $t_j$
- (4) Sending "s" command for transferring the telemetry frame to the best MP  $\text{mp}_b$ .

(iii) The following actions are performed on selected MP  $\text{mp}_b$  after receiving "s" command for transferring the telemetry frame:

- (1) Obtaining measurements data  $s_r(t_j)$  on each MP  $\text{mp}_r$ ,  $r = 1 \dots m$
- (2) Sending the telemetry frame  $s_b(t_j)$  from the best MP  $\text{mp}_b$  to the MPC (only for the best MP  $\text{mp}_b$  at time instance  $t_j$ ).

(iv) The following actions are performed at MPC:

- (1) Receiving the telemetry frame  $s_b(t_{j-1})$  from the best MP  $\text{mp}_b$
- (2) Aggregating the main stream  $s_{\text{mo}}$ .

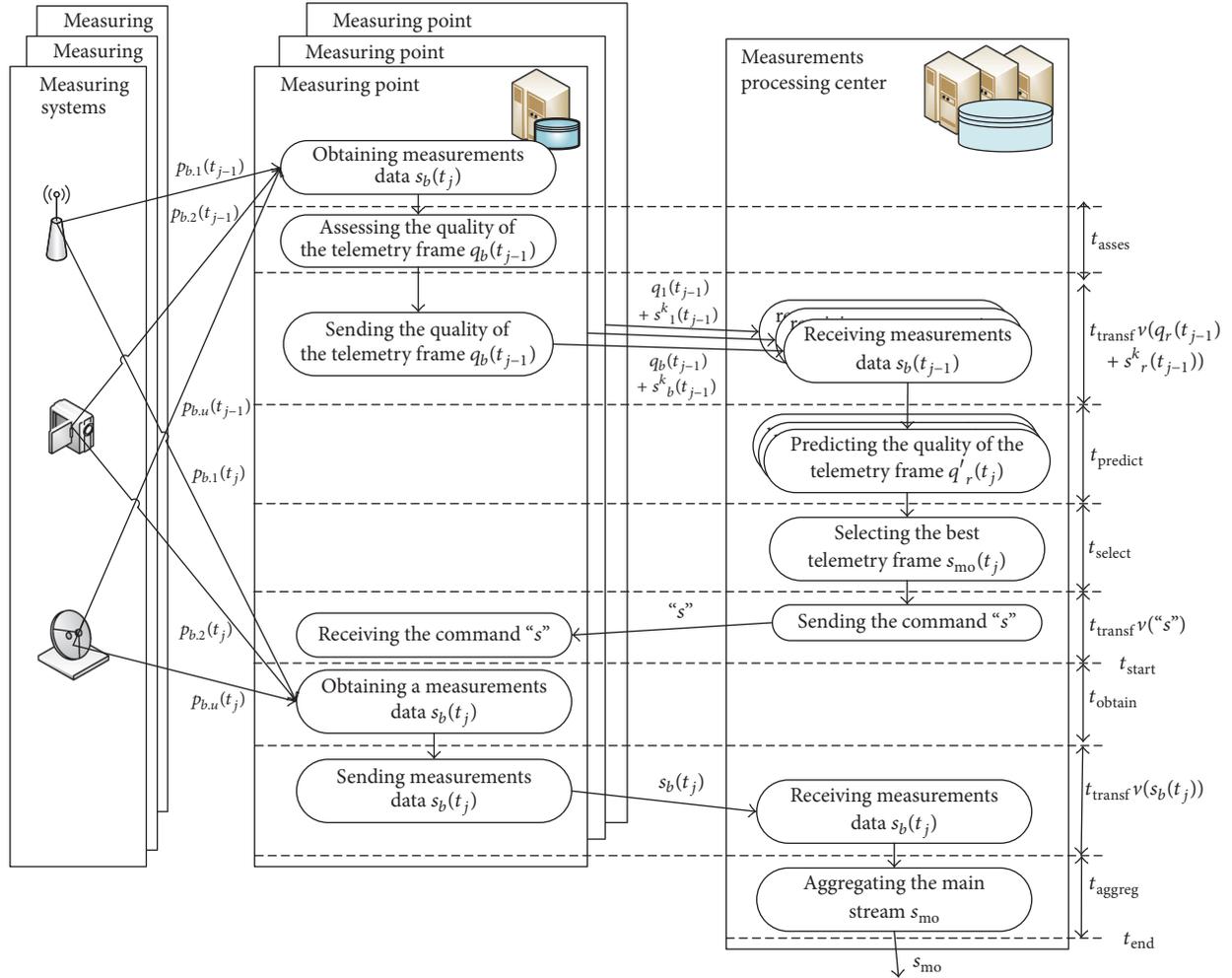


FIGURE 4: Diagram of the approach with a quality assessment prediction (QAP).

The total time for the formation of the main stream in approach QAP  $T_{QAP}$  at each instant time  $t_j$  can be calculated using the following formula:

$$T_{DQA} = \max_{r=1}^m (t_{\text{obtain}}) + t_{\text{transf}}(v("s")) + t_{\text{aggregate}}. \quad (17)$$

Time of full cycle for each measurement is calculated by the following formula:

$$\begin{aligned} T'_{DQA} = & T_{DQA} + \max_{r=1}^m (t_{\text{asses}}) \\ & + \max_{r=1}^m (t_{\text{transf}}(v(q_r^t(t_j)) + v(s_r^k(t_j)))) \\ & + \sum_{r=1}^m t_{\text{predict}} + t_{\text{select}}. \end{aligned} \quad (18)$$

In this case, network traffic  $V_{DQA}$  at each instant time  $t_j$  is determined by the following formula:

$$\begin{aligned} V_{DQA}(t_j) = & \sum_{r=1}^m (v(q_r(t_j)) + v(s_r^k(t_j))) + v("s") \\ & + v(s_b(t_j)). \end{aligned} \quad (19)$$

The quality of the telemetry frame  $q_r(t_j)$  at each time point  $t_j$  is determined by prediction accuracy of the built mining model ( $k_m$ ):

$$Q_{QAP}(t_j) = \max_{r=1}^m (q(s_r^t(t_j))) \times k_m. \quad (20)$$

Due to the fact that the volume of kinematic parameters  $s_r^k(t_j)$  is generally much smaller than the volume of telemetry frame  $s_r^t(t_j)$

$$v(s_r^k(t_j)) \ll v(s_r^t(t_j)), \quad (21)$$

this approach results in much lower network traffic than the generic approach and slightly higher network traffic than the previous one:

$$V_{DQA} < V_{QAP} \ll V_{FG}. \quad (22)$$

Also, calculations on assessment of the quality of telemetry frame are performed on each MP, which considerably relieves the MPC from the computational load.

Thus the approach QAP has low network traffic and the time of formation of the main stream. The disadvantage of

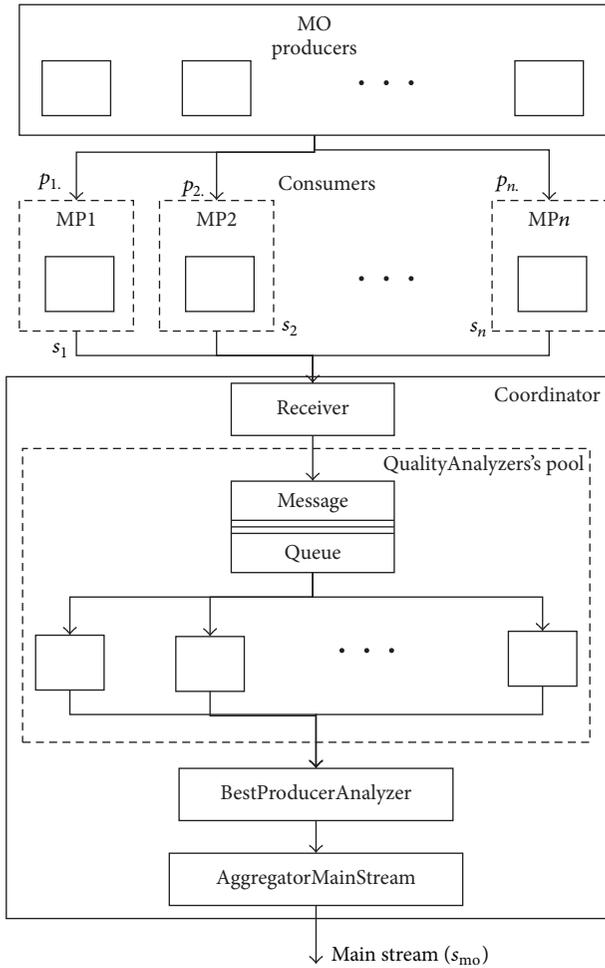


FIGURE 5: Implementation of generic approach (FG).

this approach is a potential decrease in quality of main stream. The extent of the decrease depends on the quality of the developed predictive model. However, it should be noted that even an approach with full measurement gathering fails to achieve 100% quality for all telemetry frames, as there may be time instants when all MPs receive frames with low quality. In view of this, telemetry analysis systems should provide the option of processing frames of poor quality. Therefore, the telemetry quality criterion can be noncritical.

## 4. Implementation of the Proposed Approaches

**4.1. Elements of the Measurement Data Gathering System.** In order to implement the measurement data gathering system from distributed sources, we used actors model [11] and its software implementation, Akka library [12].

The overall principle of the abovementioned mechanism using complete data transfer involving actors model is shown in Figure 5. The main entities are presented in Figure 5:

- (i) Producers (data sources) are various measurement means mounted on MO and generating the telemetric stream; each producer generates data packet with

0.25 sec interval and transfers it to the primal consumer.

- (ii) Consumers (primary consumers) are MS mounted on MP and gathering measurements coming from the sources. They form the measurement frame and transfer them to MPC for further processing.
- (iii) Coordinator (calculation center) is a set of actors hidden behind general façade and representing functionality implemented by MPC and, more specifically, preprocessing, frames quality analysis, and forming the main stream.
- (iv) Receiver is an actor presenting secondary consumer, in particular calculation center (CC). The actor is a point of interaction of the CC with all Consumers actors. After receiving another data packet, actor transfers it for processing.
- (v) QualityAnalyzers are set of actors united into pool and located in CC. They assess the quality of incoming packets. Each data packet is placed to the processing queue, upon which vacant analyzer addresses this queue and retrieves the packet for processing. Analyzed data packets are further transferred to the best producer analyzer.
- (vi) BestProducerAnalyzer is an actor determining the best snapshot among ones received from primary consumers (PCs). Analysis is performed with the packet receiving interval (0.25 sec), but there is an initial delay provided (3 sec delay was used during testing). Upon determining the best snapshot, actor transfers it to the main stream aggregator.
- (vii) AggregatorMainStream is an actor which “glues” the best snapshots into main stream. Formed stream is transferred to the end consumers. The way of implementing the main stream translation process depends on particular implementation.

**4.2. Implementation of Gathering with a Distributed Quality Assessment.** In order to implement the approach DQA, several modifications were made in the system affecting the actors model.

The QualityAnalyzers pool becomes unnecessary since the stream quality assessment is performed in MP, and there should be one actor of this type per each unit. Therefore, the functionality provided by these actors can be integrated to the Consumer-type factors. Thus, each unit shall have its own queue for snapshots received from Producers.

It is unnecessary to implement the BestProducerAnalyzer actor as an individual actor anymore, as its functionality can be combined with Receiver-type actor.

Other types of actors remain unchanged. The actors used in this implementation of the data gathering mechanism are shown in Figure 6.

**4.3. Implementation of Gathering with Quality Assessment Prediction.** In order to implement the approach with quality assessment prediction of measurements, implementation of

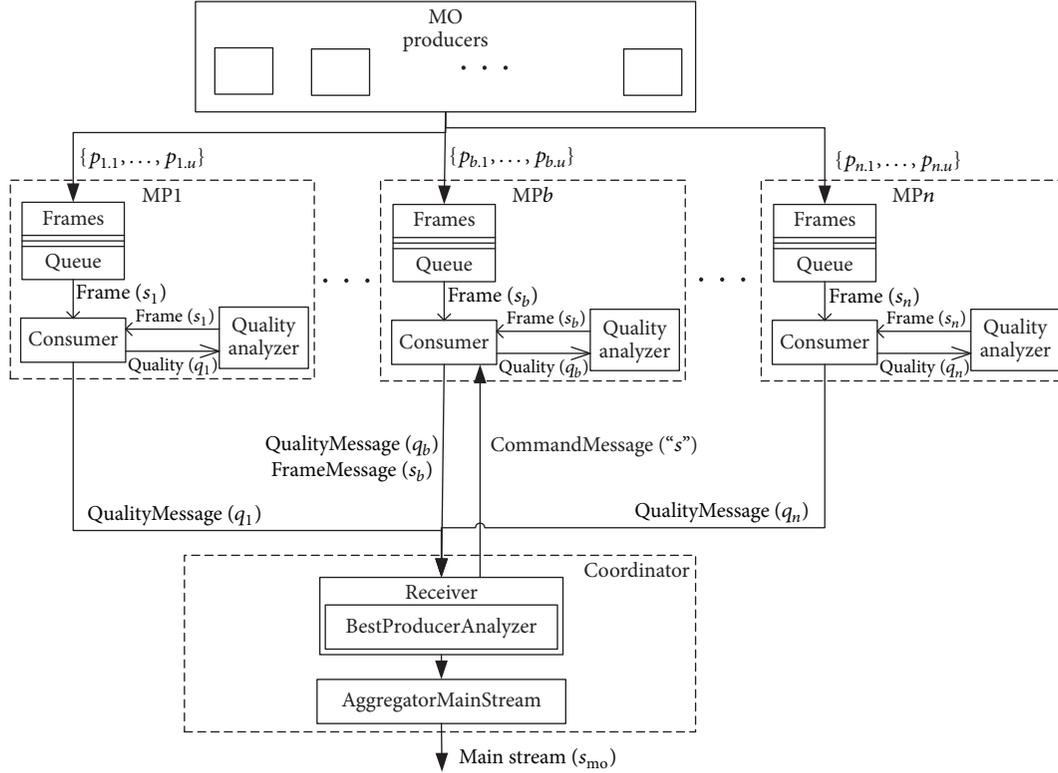


FIGURE 6: Implementation of gathering with a distributed quality assessment (DQA).

the Receiver actor was modified. It uses mining model to predict the best MP at next time instance. The mining model used for prediction was implemented as PredictionModel class. Corresponding actors are shown in Figure 7.

## 5. Experiments

Experiments were performed with the distributed system where each of the actors was located on an individual machine. Four physical computers with two virtual machines deployed on each of them were used. Ubuntu Server 16.04 was installed on virtual machines.

For experiments, we used data sets of real telemetry from flight object. The data sets were received from 2 MPs with a measuring duration of 12,5 minutes (2,700 measurements with period 0,25 sec). Each MP contains the following MS:

- (i) Radar measuring kinematic parameters: coordinates  $(x, y, z)$  and the velocity vector  $(vx, vy, vz)$
- (ii) Telemetry.

The first characteristic used to assess different implementations was the frame delay, that is, the time interval from the moment of obtaining measurements data to the moment of aggregating the main stream. Developed system allows setting the initial frame delay. This is a time interval where the start time of all system components and expected time delays in message transfer via network are set. Main stream aggregator located in CC starts the stream formation process

after the specified initial delay from the start of the test. During the test, initial delay was set as one iteration (0.25 s).

From the retrieved data, it follows that the average delay value shall be (Figure 8)

- (i) for approach with full data gathering (FG), about 0.2629 sec,
- (ii) for approach with distributed quality assessment (DQA), 0.2921 sec,
- (iii) for approach with quality assessment prediction (QAP), 0.2645 sec.

Additional delay of the distributed implementation option made by the system (except 0.25 sec of initial delay) is 0.0421 sec which is three times larger compared to the similar indication of the first implementation (0,0129 sec). Differences result from the measurements data delivery mechanisms in these two implementations. In the DQA implementation, additional delay is made at the phase of transferring the quality report to the CC and sending request for best-quality snapshot to a certain PC.

Delay in the QAP implementation is close to the result received from the generic approach. The value is increased insignificantly due to necessity to switch between data sources when swapping the best MP.

The next researched characteristic is quality of the main stream. Since the testing was performed with preformed data sets, it was possible to perform preassessment of their quality. For each data frame, quality characteristic was calculated

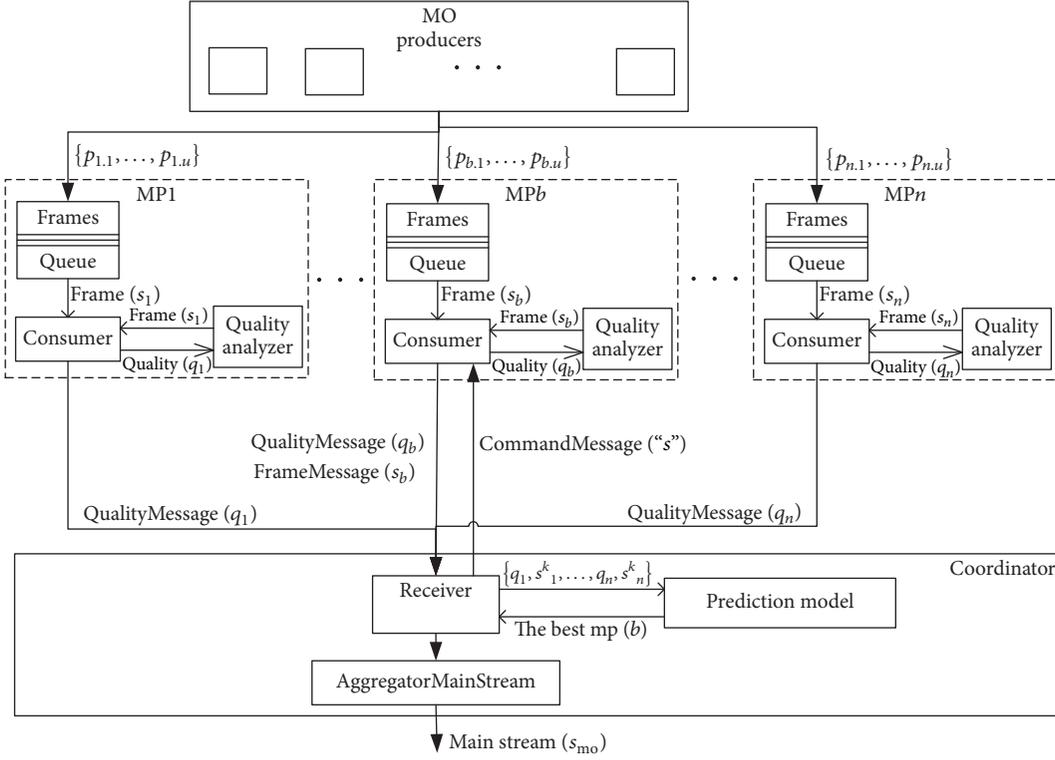


FIGURE 7: Implementation of gathering with quality assessment prediction (QAP).

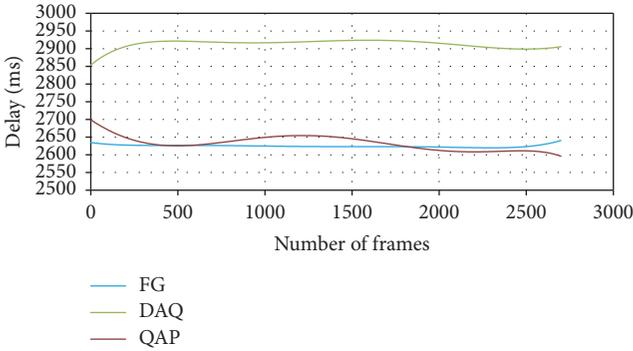


FIGURE 8: Summary diagram of the frame delay in three different implementations.

(Figure 9 for telemetry from first MP and Figure 10 for telemetry from second MP). Figure 11 shows that the quality of main stream (green line) is produced by the approaches FG and DAG.

Results received from the QAP implementation present greater interest. To predict quality, several mining models were used. They were trained by data mining algorithms from the Weka library [13]:

- (i) GaussianProcesses [14]
- (ii) MultilayerPerceptron [15]
- (iii) RegressionByDiscretization [16]

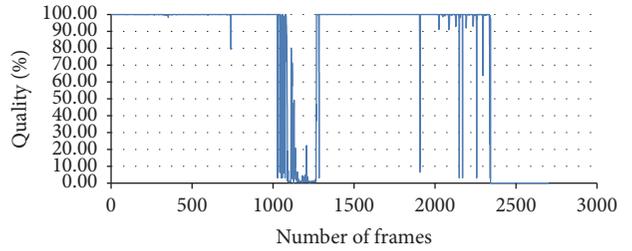


FIGURE 9: Diagram of the frame quality assessment from the first MP.

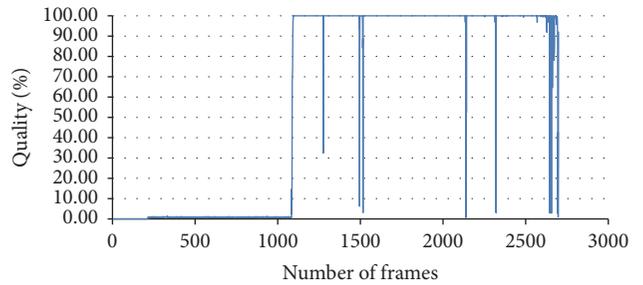


FIGURE 10: Diagram of the frame quality assessment from the second MP.

- (iv) Decision tree (M5PBase) [17]
- (v) SVM regression (SMOreg) [18].

The following metrics were used to assess the accuracy of the prediction model (where  $y$  is a true value,  $y'$  is a value

TABLE I: Comparison of prediction models.

Prediction model	Correlation coefficient	MAE	RMSE	RAE	RRSE
GaussianProcesses	0,3977	15,9667	25,3496	100,90%	91,71%
MultilayerPerceptron	0,9562	4,034	8,2296	25,49%	29,77%
RegressionByDiscretization	0,9922	0,3714	3,4396	2,35%	12,44%
SMOreg	0,4246	8,8983	28,9597	56,23%	104,77%
M5P	0,989	1,5178	4,4831	9,59%	16,22%

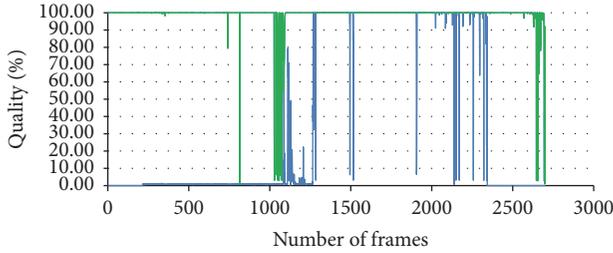


FIGURE 11: Diagram of the main stream (green color) quality assessment for the FG and DAG implementation.

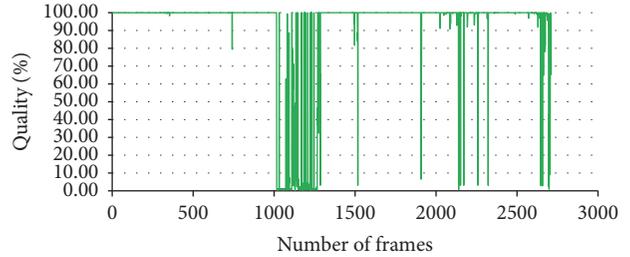


FIGURE 12: Diagram of the main stream assessment using RegressionPredictionModel.

estimated using some algorithm, and  $\bar{y}$  is a mean value of  $y$ ):

(i) Mean absolute error (MAE):

$$\text{MAE} = \frac{\sum_{i=1}^n |y'_i - y_i|}{n} \quad (23)$$

(ii) Root mean squared error (RMSE):

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (y'_i - y_i)^2}{n}} \quad (24)$$

(iii) Relative absolute error (RAE):

$$\text{RAE} = \frac{\sum_{i=1}^n |y'_i - y_i|}{\sum_{i=1}^n |\bar{y}_i - y_i|} \quad (25)$$

(iv) Root relative squared error (RRSE):

$$\text{RRSE} = \sqrt{\frac{\sum_{i=1}^n (y'_i - y_i)^2}{\sum_{i=1}^n (\bar{y}_i - y_i)^2}}. \quad (26)$$

The results of the comparison are presented in Table 1.

The prediction models MP5 and RegressionByDiscretization have the best metrics. They were selected for the implemented approach of gathering with quality assessment prediction.

As a result of using the MP5 model, the results were close to the one presented in Figure 11. The correlation coefficient of these two series was 0.9844, which conforms with theoretical assessment presented in Table 1. The data received from RegressionByDiscretization testing are slightly different from

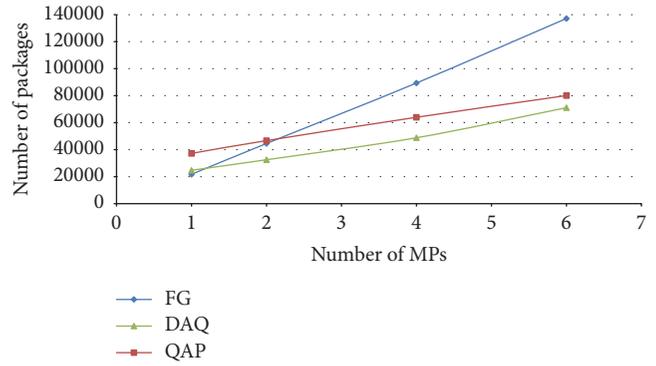


FIGURE 13: Summary diagram of packet quantity dependence from quantity of PCs.

theoretically predicted ones (Figure 12). In this case, the correlation coefficient was 0,7841.

As it can be noticed, the model fails in the areas where the best sources swap. This may indicate incorrectly calculated dependence between parameters for prediction model.

Another criterion for implementation comparison is network traffic volume analysis. Measurements were made with Wireshark utility [19], a traffic analyzing software for Ethernet network.

Traffic measurement was made with different system configurations, in particular with varying PC quantity used in test emulations. In this kind of test, quality of the formed main stream was not taken into account. Thus, for each implementation variant, 4 system configurations were prepared: with 1, 2, 4, and 6 PCs. Test was performed three times for each configuration, and results were averaged and generated traffic volume was assessed based on those values.

Diagrams (Figures 13 and 14) represent results of the implementation testing. The size of one source data file is

TABLE 2: Experimental results.

Approach	Delay (second)	Mining model correlation	Number of MPs	Network traffic (Mb)	Quality of main stream
Generic approach	0.2629	-	2	686	93%
			4	1,376	
			6	2,077	
Approach with distributed quality assessment	0.2921		2	374	93%
			4	396	
			6	412	
Approach with quality assessment prediction by regression model	0.2645	0,9807	2	375	83%
			4	401	
			6	420	
Approach with quality assessment prediction by M5PBase model		0,9526	2	372	92%
			4	399	
			6	419	

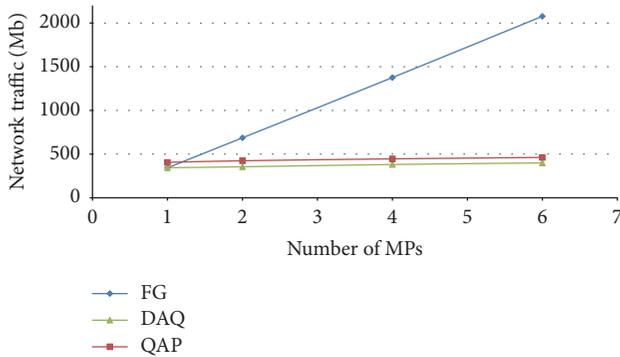


FIGURE 14: Summary diagram of traffic volume dependence from quantity of PCs.

~339 Mb. It can be seen that the dependence of the traffic volume from quantity of MPs is linear, wherein actual data confirm theoretical assumptions about linear dependence of the traffic volume from source data volume and, at the same time, number of MPs in the system.

Analysis of the above data allows concluding that the implementations of proposed approaches use much greater number of TCP packets for data transfer (37241 compared to 21752 with a single MP). Yet, specific size of these packets is significantly smaller, as the general traffic volume for one used MP is 355.31 Mb compared to 342.715 Mb in the implementation of generic approach. This can be explained by smaller size of the packets formed for transferring. At the same time, traffic volume for implementation of proposed approaches is not directly dependant on quantity of MPs in the system. This significantly reduces total volume with their greater amount (411.98 Mb compared to 2077.23 Mb with 6 active MPs).

Table 2 summarises experimental results for all approaches of measurement data gathering.

The experimental results showed a significant reduction of traffic transmitted over the network by using approaches to distributed measurement data gathering. However, the

DAQ approach had a bigger delay to form the main stream. Using prediction model for quality assessment solves this problem. At the same time, the quality of a main stream varies depending on the selected mining model used to predict the stream quality.

## 6. Conclusion

Use of Fog computing for data gathering from moving objects allows not only reducing the load to data processing center by putting part of calculations on MP, but also significantly reducing the network traffic. Both approaches give significant reduction of the network load. Also, when increasing quantity of MPs in these variants, network traffic does not increase, which allows increasing the number of measurement tools in the system and gaining better control over the objects. This allows using wireless networks with limited capacity for data gathering.

The disadvantage of the distributed quality assessment approach is an increase of time delay in the main stream forming. If this value is crucial, it is possible to use approach with assessment prediction. It gives the same time delay as the implementation with full data gathering.

Quality of the formed main stream with this approach is fully dependent on prediction model implementation. Prediction can be improved also by model learning in the process of measurement values gathering. It is the objective of future research.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This paper has been prepared with the support of the Ministry of Education and Science of the Russian Federation in the framework of the state work "Organization of Scientific Research" of State Task no. 2.6113.2017/BY.

## References

- [1] G. Miao, J. Zander, K. W. Sung, and S. Ben Slimane, *Fundamentals of Mobile Data Networks*, Cambridge University Press, Cambridge, UK, 2016.
- [2] Gartner, “Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020,” STAMFORD, Conn., 2013, <https://www.gartner.com/newsroom/id/2636073>.
- [3] C.-W. Tsai, C.-F. Lai, and A. V. Vasilakos, “Future internet of things: open issues and challenges,” *Wireless Networks*, vol. 20, no. 8, pp. 2201–2217, 2014.
- [4] F. Chen, P. Deng, J. Wan, D. Zhang, A. V. Vasilakos, and X. Rong, “Data mining for the internet of things: Literature review and challenges,” *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 431047, 2015.
- [5] S. Bin, L. Yuan, and W. Xiaoyi, “Research on data mining models for the internet of things,” in *Proceedings of the 2nd International Conference on Image Analysis and Signal Processing (IASP’10)*, pp. 127–132, April 2010.
- [6] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswamia, “Rajkumar Buyya, Slaven Marusic, Marimuthu Palaniswamia, Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions,” *Future Generation Computer Systems, Elsevier, Volume 29, Issue*, vol. 29, no. 7, pp. 1645–1660, 2013.
- [7] A. El-Sharkawi, A. Shouman, and S. Lasheen, “Service Oriented Architecture for Remote Sensing Satellite Telemetry Data Implemented on Cloud Computing,” *International Journal of Information Technology and Computer Science*, 2013, Published Online June 2013 in MECS.
- [8] C. Mirchandani, “Cloud-based Ground System for Telemetry Processing,” in *Proceedings of the Complex Adaptive Systems, 2015*, pp. 183–190, USA, November 2015.
- [9] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in *Proceedings of the 1st ACM Mobile Cloud Computing Workshop, MCC 2012*, pp. 13–15, Finland, August 2012.
- [10] R. W. Hockney, “The communication challenge for MPP: Intel Paragon and Meiko CS-2,” *Parallel Computing*, vol. 20, no. 3, pp. 389–398, 1994.
- [11] C. Hewitt, P. Bishop, and R. Steiger, “A universal modular ACTOR formalism for artificial intelligence,” *International Joint Conferences on Artificial Intelligence*, pp. 235–245, 1973.
- [12] “Akka Documentation,” <http://akka.io/docs/>.
- [13] I. H. Witten, F. Eibe, and A. Mark, *Data Mining: Practical Machine Learning Tools and Techniques*, vol. 629, Morgan Kaufmann Publishers, San Francisco, 3rd edition, 2011.
- [14] C. E. Rasmussen, “Gaussian processes for machine learning,” in *Advanced Lectures on Machine Learning*, vol. 3176 of *Lecture Notes in Computer Science*, pp. 63–71, The MIT Press, Cambridge, Mass, USA, 2004.
- [15] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer series in statistics, Springer, New York, NY, USA, 2008.
- [16] D. D. Rucker, B. B. McShane, and K. J. Preacher, “A researcher’s guide to regression, discretization, and median splits of continuous variables,” *Journal of Consumer Psychology*, vol. 25, no. 4, pp. 666–678, 2015.
- [17] J. R. Quinlan, *Learning with continuous classes*, University of Sydney, Australia, 2006.
- [18] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, and K. R. K. Murthy, “Improvements to the SMO algorithm for SVM regression,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 11, no. 5, pp. 1188–1193, 2000.
- [19] “Wireshark Developer’s Guide,” <https://www.wireshark.org/docs/wsdg.html.chunked/>.



**Hindawi**

Submit your manuscripts at  
<https://www.hindawi.com>

