



OPEN ACCESS

EDITED BY
Lianbo Ma,
Northeastern University, China

REVIEWED BY
Tian Zhang,
Northeastern University, China
Nan Li,
Northeastern University, China

*CORRESPONDENCE
Dong Li,
lidong@sia.cn

SPECIALTY SECTION
This article was submitted
to Smart Grids,
a section of the journal
Frontiers in Energy Research

RECEIVED 27 June 2022
ACCEPTED 14 July 2022
PUBLISHED 27 September 2022

CITATION
Chen D, Li D and Guo R (2022), Design
of topic priority based on industrial
publish-subscribe system.
Front. Energy Res. 10:979174.
doi: 10.3389/fenrg.2022.979174

COPYRIGHT
© 2022 Chen, Li and Guo. This is an
open-access article distributed under
the terms of the [Creative Commons
Attribution License \(CC BY\)](#). The use,
distribution or reproduction in other
forums is permitted, provided the
original author(s) and the copyright
owner(s) are credited and that the
original publication in this journal is
cited, in accordance with accepted
academic practice. No use, distribution
or reproduction is permitted which does
not comply with these terms.

Design of topic priority based on industrial publish-subscribe system

Dejian Chen^{1,2,3}, Dong Li^{2,3*} and Renchun Guo¹

¹College of Information Engineering, Shenyang University of Chemical Technology, Shenyang, China, ²Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, ³Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang, China

As the size of the industrial systems increases, more and more topic messages need to be published and subscribed to in industrial scenarios such as microgrids and manufacturing industries. A priority scheduling algorithm based on the hotness of message topics is proposed to address the problem of efficient distribution of messages with different topics in microgrids. It is an algorithm that gives different priority permissions according to the number of topics, so that in the face of a large number of different topic messages, the data messages with a higher number or urgency can be transmitted first, solving the problem of port blocking when the server distributes the messages. At the same time, to prevent low priority data from sending timeout or waiting for a long time, a time period is set up, a low priority transmission mechanism is set up within a specific time period to ensure that the system can send data to customers in a timely and accurate manner. The results show that the priority design in this paper can well solve the power scheduling problem of multi-target data transmission and ensure the transmission efficiency of messages in the microgrid system.

KEYWORDS

published and subscribed, microgrids, priority, time period, multi-target data

Introduction

The publish-subscribe model is widely used in industrial systems such as microgrids, manufacturing, etc. The microgrid is a small power system consisting of distributed power sources, loads, energy storage, power distribution, and control systems. Compared with the large grid system, the microgrid has less cost and fewer emissions and can achieve the advantages of self-control, self-management, and protection. The microgrid can be connected to and disconnected from the external grid. The microgrid is a kind of mini-grid that collects all kinds of energy facilities and all kinds of electronic power equipment, which solves the problem of grid-connected operation of large-scale power grids from local. Microgrid technology is becoming more and more mature in application engineering (Feng et al., 2011; Qin et al., 2012; Yu et al., 2015), with safe and stable operation, and the microgrid system is gradually commercialized. With the continuous development of the smart grid, the security management of power system data communication becomes more and more important (Ericsson, 2010). Large amounts

of renewable energy and smart power devices are connected to the grid, and the growing size of the grid makes the system increasingly complex. And customers' demand for power quality is gradually increasing. This makes the real-time data and system operability of the smart grid gradually improved.

In the smart grid system, the publish-subscribe system can be well used in the communication between the underlying equipment and the upper-level data and is suitable for large-scale application scenarios. The publish-subscribe model can be applied to Web Services (Hemayati et al., 2010; Sun et al., 2016; Cheng et al., 2018) to improve its proactivity. The publish-subscribe model has great advantages in the decoupling of space and time (Zhai et al., 2008). Publishers and subscribers do not need to be online at the same time, and the publish-subscribe function can be realized at any time. Publishers are responsible for passing messages to the server, and subscribers are only responsible for subscribing to messages of interest to them (Feng et al., 2008). Currently, to solve the problem of publish-subscribe efficiency, many methods have been proposed. Two types of parallel matching algorithms are proposed, and with the increase of threads, the matching time will be greatly reduced (Qian et al., 2011). The word linked list uses the predicate filtering ability, the larger the predicate value range, the higher the priority (Xu et al., 2013). The attributes in the attribute space are combined, and the event matching process is only executed on some relevant units of the index structure, which reduces the number of subscriptions involved in the event matching process (Yang et al., 2016). Searching according to the corresponding id order list is also a way to correspond to a large-scale publish-subscribe system (Xu et al., 2014). QuTREE is a new tree matching structure, combined with the IDAGSM algorithm to divide the interval to reduce the depth of the tree and improve the matching efficiency (Qiao et al., 2015). GEM (Geometric Event Matching) improves the matching performance of events in a manner similar to the triangular index structure (Fan et al., 2016), etc.

The publish-subscribe system is often used in communication systems with a large amount of data, while the capacity of the server to process messages is limited, and the capacity of message queues and the size of bandwidth are fixed. Therefore, it is also important to know how to solve the problem of publish-subscribe network congestion. Reorganizing the queue of filter-covered subscriptions can improve the metric performance of the queue (Yan et al., 2013). The marginal revenue-based scheduling (MRBS) algorithm can prioritize important messages and avoid performance instability when the system is overloaded (Yan et al., 2010). The multi-objective optimization can effectively relieve the tension of the system. For the resource allocation problem in constrained MEC, a double auction (TCDA) mechanism with a dynamic pricing strategy is proposed to effectively implement resource allocation and pricing to achieve authenticity and budget balance (Ma et al., 2021a). An adaptive

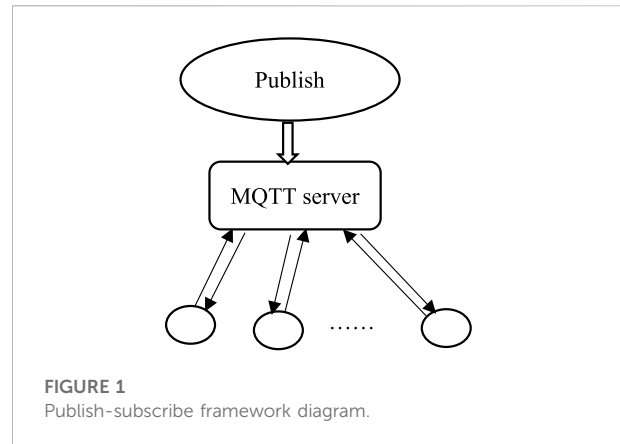


FIGURE 1
Publish-subscribe framework diagram.

reference vector reinforcement learning method can be applied to solve multi-objective optimization problems and is effectively used to search for the best recipe for raw materials in the copper smelting process (Ma et al., 2021b). According to the guidance of the reference vector, the adaptive strategy is used to optimize the decision variables, which can effectively solve the problem of large-scale multi-objective optimization (Ma et al., 2021c). Using the user search experience of the two OD engines can optimize the search direction, which has a good role in optimizing complex functions (Ma et al., 2021d).

The heat priority algorithm designed in this paper is a matching algorithm that prioritizes topics based on the number of subscribed topics, and the server gives priority to different topics and distributes topics according to the priority sequence. The maximum number of subscriptions can be processed in the shortest possible time, effectively solving the problem of network message congestion. If the number of subscriptions is large enough, to prevent the low priority from being unable to receive messages for a long time, a time feedback mechanism is set up for the low priority, so that the low priority and the server maintain the echo of the time period. The experimental results demonstrate the superiority of the prioritization and feedback mechanism.

The specific workflow in Figure 1 is as follows:

- a) After the basic equipment completes the tasks of data collection, transformation, collection, and processing, it sends the collected data to the MQTT server as the publisher. The MQTT server stores these data in some nodes.
- b) If the upper-level database wants to get these data, it needs to subscribe to the MQTT server as a subscriber. After the database receives the data, it needs to compare it with the standard data in the database. At this time, the information obtained from the comparison will be sent to the MQTT server as the publisher. The MQTT server stores the data.

- c) The underlying device needs to get the comparison data from the database, needs to subscribe to the MQTT server as a subscriber, and adjust the parameters according to the comparison message in time after receiving the data.
- d) The same goes for device-to-device communication. If the data volume is too large for the MQTT server to complete the data distribution task in time, then the subscriber will keep sending subscription requests to the MQTT server until the MQTT server completes the data distribution.

Related theory

The publish-subscribe system of this article applies the MQTT protocol (MQTT, 2020). The MQTT protocol is very standard. The advantages of the MQTT protocol are that the first is easy to apply, the protocol is relatively complete in all aspects, and communication can be achieved after applying the protocol on both the publishing and subscribing ends. The second protocol has a large open-source program, combined with the existing Internet technology, making it increasingly easy to develop. The third interface presents standardized, third-party customers or devices that want to connect to the platform, only need to combine a standardized set of MQTT protocol interfaces can be, so that the multi-party platform communication becomes simple. MQTT has a wide range of applications (Chi et al., 2020; Ganguly and Chatterjee, 2020; Huang et al., 2021; Yang et al., 2021).

The MQTT protocol comes with a heartbeat mechanism, which does not need to be set up separately, and also includes testamentary messages, asynchronous mechanisms, etc., which can be well used in the Internet of Things, so the MQTT protocol is suitable for many scenarios. For example, some APP smart devices control end applications, network chat, end-to-end communication, etc. The protocol is especially widely used in industrial environments, which can easily make users and IoT connect and interact with information. Currently, the MQTT protocol can provide more flexible and reliable messaging services in IoT systems (Wu and Li, 2018). The combination of the MQTT protocol and the intelligent gateway has a good application prospect (Lima et al., 2019). The MQTT protocol has also been applied to message push (Kumar and Johari, 2015) and remote real-time communication (Ding et al., 2016) on different platforms.

In today's Internet age, QoS (Quality of Server) (Behnel et al., 2006) plays a vital role. QoS can set up different quality of service for users in need so that the main performance indicators such as data throughput, delay, and packet loss rate during network transmission are well improved. There are three levels of QoS, and different load devices applying different QoS levels can build a good network environment (Lee et al., 2013). The application of

QoS can improve the reliability of the system and the timeliness of achieving user requirements (Zhang et al., 2015). MQTT itself has a reliable mechanism to ensure message transmission. It not only designs message response, data storage, and data retransmission but also has three levels of QoS to ensure reliable message transmission. The QoS used in MQTT of this system is QoS0, and the connection performance between the publish-subscribe system is stable. The subscription side can be accessed several times to achieve the stability of the system reception. MQTT Server is the key to the priority design of this paper. MQTT Server can be well used in publish-subscribe systems and carries the task of data distribution very well.

The design of this paper has the following improvements from the traditional priority. First, the traditional priority service is a fixed first-come-first-served model. The priority of this paper can be changed on the server-side according to the subscription topic, which can be well adapted to different environments and is more flexible than the traditional priority. Second, the traditional priority for the low-priority processing method is usually priority adjustment, this paper set up a time period, every period of time for low-priority sexual information interaction, the time period can be changed according to the needs of subscribers. Third, to ensure the stability of the system, the server will adjust the publish-subscribe of messages and the frequency of data distribution. Fourth, the server-side can set the level of the subject sequence, and in the case of the same priority, the successive publish-subscribe can be adjusted according to the depth of the data topic.

Publish-subscribe system framework and priority design

Publish-subscribe system framework

In the microgrid system environment, the underlying equipment often needs to exchange information in time with the upper-level database. The proxy server connects the upper-level database and many underlying devices and is an important part of the publish-subscribe system framework. Treating the presence of the underlying device and the upper layer database as multiple communication devices. These communicating devices can act as both publishers and subscribers so that devices can communicate with each other as long as there is the same communication protocol. The device can be connected to the server by IP and port. The server acts as the data-carrying part of the whole architecture and takes on the task of data publishing and message matching. The topics that publishers want to publish and the topics that subscribers need to subscribe to are matched one by one on the server. Therefore, the communication between devices or the communication between devices and databases needs to use a server to ensure data matching and distribution.

The publish-subscribe priority design process is as follows

- a) When the underlying devices communicate too much data with the upper layer data, which exceeds the distribution capacity of the server, the communication between devices will be hindered, and the time of publishing and subscribing will be delayed, which has a great impact on the intelligent grid.
- b) At this point, the server will establish a priority level for a period of time-based on the frequency of information exchange topics between devices, and the data represented by the topic messages with high priority can be prioritized for the execution of publish-subscribe tasks, while the data with a low priority level has to wait for the completion of the transmission with a high priority level before execution.
- c) High priority enjoys all resources, which ensures transmission efficiency. The server processes node information and processes it according to the priority list. Handling message data in a priority list solves the problem of port congestion.
- d) If the amount of high-priority data is too large or the priority list is too long, some low-priority messages will time out because the messages cannot receive messages for a long time. Setting time period feedback adjustment for this situation. If an individual message times out, the message will be distributed within the time period.
- e) The adjustment of the feedback time period will not have much effect on high-priority messages. If the lower priority needs to communicate with database data in real-time, it can be done by setting the corresponding time period. The messages are distributed in an orderly manner according to the timeout of the low priority level.
- f) If an individual device needs to communicate with other devices due to an emergency, it needs to be fed back to the server, and the server will adjust the message to a high priority to ensure the transmission efficiency of the microgrid system.

Publish-subscribe stage process

In the publish-subscribe phase, each client can both publish and subscribe to messages. The establishment of each client first needs to be initialized, and then connect to the server through IP and port number, and authenticate with the server through authentication. Turn on listening mode after the connection is completed to detect messages published and subscribed by the server.

Subscriber Subscription Section

- a) The subscriber is responsible for sending the subscribed topic messages to the server. The server is responsible for counting the number of subscription topics and integrating them and then giving the subscription topic a priority. Subscribers can

generate subscription messages within a certain time interval, and set their timeout period according to their own needs. The specific steps are as follows:

- a. When subscribers subscribe to a topic, first determine whether there is already a subscription for them. If there are duplicate subscriptions, jump out of the subscription collection directly.
 - b. Join the subscription collection, Create a subscription list.
 - c. Traverse the subscription collection and add the subscription element to the subscription list if the element appears for the first time.
 - d. Package and send subscription information.
- b) After the subscribed topic message arrives at the server, the server matches and distributes the message topic according to the priority sequence. If the match is successful, the server will package the topic message that the subscriber wants and send it to the subscriber. When the subscriber receives the message, the subscription part is also completed.

The publisher publishes the topic section:

In the message topic publishing stage, the publisher will publish the topic messages that need to be published to the server, and then these messages will enter the server. The server will match these messages with the topic messages subscribed by the desired subscribers and complete the distribution. The specific process is as follows:

- a) The publisher judges the QoS level and discards it if it does not meet the requirements of this system, and then configures the payload, messageid, and QoS. After finishing, it is packaged and sent to the server.
- b) The publisher will generate a time interval to publish the desired topic and content to the server. These topic messages will be stored in the cluster nodes in the server. If there are subscribers subscribed to the message, the topic messages will be matched with the subscriber's subscription topic content one by one. If the matching content meets the requirements for publishing, the server will arrange the priority sequence to complete the publishing of these topic messages.
- c) If the content of all messages published by publishers in a stage reaches the subscribers, the task of this publishing stage is completed.

MQTT server part and algorithm

MQTT server part

The MQTT server uses TCP/IP to connect to the client, and the main functions include message publishing and subscribing, disconnection and reconnection, message monitoring, logging, etc. When the MQTT server starts to work, it will continuously listen for messages from all clients,

and it will continue to perform matching and distribution tasks after listening to the messages. If the client is disconnected, the message listening for this message will fail, and the MQTT server will stop the distribution task of the message. The MQTT server is mainly responsible for data connection and data forwarding. The MQTT server protocol is responsible for the message communication between the device and the server. The API interface function module of the MQTT server is responsible for calling different programs. The MQTT server is also accompanied by heartbeat packets to maintain continuous connection capabilities. Different users need authentication when connecting to the MQTT server, which adds multiple guarantees to the security of the server. The MQTT server has a good effect in the application scenarios of large-scale data packets (Yue et al., 2017).

The MQTT server is the core part of the publish-subscribe, and the part that is particularly vulnerable to message publish-subscribe hindrance. Therefore, the design of priorities in this section is particularly important. This part is connected with many devices for IP and port, and the connection is safe and reliable to ensure the stability of data transmission. The MQTT server carries the task of matching and distributing a large amount of data. The server connects to various devices through ports and listens to messages from various subscribers. Prioritizing all topic messages can solve the problem of time delay very well, which is also the focus of this paper. Setting up a priority service for the MQTT server not only subscribes topic messages to the subscriber but also effectively distributes the messages published by the publisher, to alleviate port congestion to the greatest extent.

If there are N levels of priority messages on the server within a certain period of time, the N topic messages are regarded as 1, 2, ..., n respectively. The priority of N topic messages is remembered as P_1, P_2, \dots, P_n , the larger the number, the higher the priority. The server distributes the priority sequence in order from largest to smallest. The weight of the message is denoted as W , W_i represents the weight of the i -th topic message. A topic message with a high priority will have a higher weight than other topic messages. If an urgent message occurs, it is marked as U , indicating that the message needs urgent processing. The average publication and subscription delay of each message is denoted as D , D_i represents the message delay of message topic i . d_x represents the topic depth of message x . The waiting time WT for the server to distribute the message, WT_i represents the distribution waiting time of the i -th topic message, The delay time DT for the subscriber to receive the message, DT_i represents the delay time for the subscriber to receive the message of the i -th topic without blocking.

For the determination of network blockage, the i -th topic message is used to represent (Ying-Ying, 2014).

$$\frac{WT_i}{DT_i} > 0.5$$

This situation means that the waiting time for the server to distribute the queue message exceeds half of the delay of the subscription message, which means that the server cannot distribute the data packets in time, and there is a situation of network congestion. A larger ratio means more severe network congestion. If the distribution and subscription of messages are carried out in the case of a large ratio, the efficiency of the entire system will be greatly affected. At this time, a good scheduling mechanism is required to match and schedule the subscription messages of the server.

In the case of network congestion, for any given i, j

$$\text{if } P_i > P_j, \text{ then } W_i > W_j, D_i < D_j$$

Prove that the priority of topic message i is higher than that of topic message j , then the weights of topic message i and topic message j will be arranged according to the priority. The delay of topic message i will be smaller than that of topic message j , and the topic message i will preferentially preempt resources to publish and subscribe messages.

$$\text{If } \exists U, \text{ then } P_U > P_{max}$$

If the topic message that appears is an urgent message U , the server will adjust the priority of the topic message upwards. The emergency message U can determine the sequence of priorities without calculating the number by the counting method, and can directly distribute the topic message in priority, which ensures the distribution efficiency of the emergency message.

If topic message i and topic message j have the same priority, a judgment is made according to the following:

$$\begin{aligned} \text{If } \exists P_i == P_j, \\ \text{if } d_i < d_j, \quad \text{then } P_i > P_j, \\ \text{else contrary.} \end{aligned}$$

If there are two topic messages with the same priority, it proves that the number of subscriptions to these two topic messages is the same, and then the weights of the two topic messages are the same. At this time, if the two topic messages need to be prioritized, the priority level can be determined according to the depth of the topic. The complexity of matching data with the server is also higher for topics with higher depth, which can be prioritized with topics with shallow depth. In this way, the server can complete the data matching at the fastest speed and distribute the data to the desired customers in time.

Matching algorithm

The MQTT server integrates the received subscription topics and content. Then, the subscribed QoS service, topic level, and data content are recorded separately, and the required data is sent to the subscriber according to the path of the IP and port of both parties. For the topic messages that need to be published to be recorded. The server extracts the required data for matching. If it is detected that there are other subscribers subscribed to the topic message, the information is matched one by one. Subvalidator has a subscription validation function. Topicfilter has a filter function.

Subscription Matching:

```
Void Subprocess(message) { // Calibrate-Subscribe
  For mqttSub: message.payload().topicSubscriptions()
  {If (Subvalidator == NULL && topicfilter)
  {GrantQosList.add(mqttSubQos);
  SubTopicList.add(mqttSubTopic);
  SessionManage.addsub (mqttSubTopic, clientId,
  mqttSubQos.value());}}
```

If the validator is working correctly and the filter has filtered out the useless messages, the remaining messages are validated. All subscription information is recorded.

Prioritization algorithm

In a certain period of time, the server will integrate different topics subscribed by subscribers, and sort them according to the number of different topics. Using a count to calculate the number of topics, the larger number will get higher priority and can enjoy the right of priority distribution, while the smaller number of subscription information will get the lower priority sorting level. For urgent messages, processing can be given a high priority sequence without sorting. Topic_info indicates the information on the topic. Period is set to a period.

When receive subscribe topic set

For each topic

if receive topic_info, count++;

Sort count list;

If count has a maximum value

Mqttserver(topic_info);

Count = 0;

If Period == 0, return; //If the time runs out, the operation ends within the time period

The period begins when the server receives data and continues until the end of the running period when it reaches 0, indicating the end of a period. If the server receives additional multiple subscriptions again, the Period is retimed and the server collects and prioritizes the information again.

Get the priority algorithm according to the number of different topics calculated by count, and give the corresponding priority level. The server will assign the order of publish-subscribe according to the priority level. The time for

high priority to perform data distribution is set to T. The server will determine if there is a low priority that needs to publish-subscribe to the message due to timeout within T time.

Void Mqttserver (topic_info)

When publish-subscribe topics

while time \in T, continue;

If \exists timeout //Determine if there is a low-priority message with a timeout, Time period setting;

publish-subscribe (topic_info); //Publish and subscribe information

if T == 0, return;

The server gives priority to high-priority messages and timeout messages during T time. If T equals 0 it means that the high priority messages are finished running and then the other priority messages are executed.

Low-priority feedback adjustment mechanism

This paper is designed in an industrial intelligent environment. The information exchange between devices and devices and between devices and databases is often frequent, and the amount of data is relatively large. Priority can solve the problem of port congestion well, but the distribution of low-priority messages may be hindered. Therefore, a feedback mechanism for low priority is designed. Feedback adjustment is an adjustment mechanism designed for individual low-priority subscribers who cannot receive messages for a long time. Usually, to prevent this from happening, considering weight conversion, priority readjustment, which will increase additional traffic needs, or allocating other resources is also an effective solution (Merayo et al., 2017). In this paper, using the time period to regularly send low-priority messages within limited resources and without a timeout. A low-priority subscriber will send a request to the server after a long period of time without a subscribed message, and the server will give that low-priority message a time period to guarantee the transmission of its message. WT means the message waits for the time. DT represents the delay time for the subscriber to receive the message. High-Low priority (High priority, Low priority).

For Low priority subscribers

If WT == DT //When the wait time is approaching the timeout

Time period adjust //Different time periods and

subscription frequencies can be set publish(Low priority);

If High priority ends

publishAll(Low priority);

continue;

For timeout messages adjust the time period and distribution frequency to ensure that the message will not time out. If the distribution of high priority messages is complete, all remaining low priority messages will be distributed until the distribution of low priority messages is complete.

Adjust the distribution frequency, F_i represents the distribution frequency of topic message i

$$\text{if } F_i > F_j, \text{ then } W_i > W_j$$

$$\text{if } F_i > F_j, \text{ then } D_i < D_j$$

If there is a long queue of high-priority messages and a large number of messages, it is difficult for low-priority messages to be distributed in time. At this time the server can adjust the distribution frequency for low priority in the time period, by changing the distribution frequency of messages can complete the distribution of messages for low priority messages in a timely manner. The change in distribution frequency does not affect the priority sequence, but it changes the weight of the distribution frequency messages, and the change in weight makes some low-priority messages available with faster distribution. Similarly, the change in distribution frequency causes the delay of the message to change as well.

The distribution frequency can be adjusted accordingly to the needs of the subscribers. If the subscriber wants a more time-sensitive message, it can be distributed with an increased frequency of messages. If the subscriber requires high transmission quality and message delivery rate, then the server can set a low frequency for message distribution. Just make sure the message doesn't time out. The server sets up the different frequencies of message distribution for low-priority messages according to the requirements of different subscribers. In this paper, the system can save the data in the server node, and match the subscribed topic messages according to the priority sequence, which ensures the transmission quality of the messages. For subscribers with different needs, different distribution frequencies can be set to ensure distribution efficiency.

Experimental environment

The programming environment of this paper is in the IEDA compiler of the Windows system, and a computer is configured for experimental analysis.

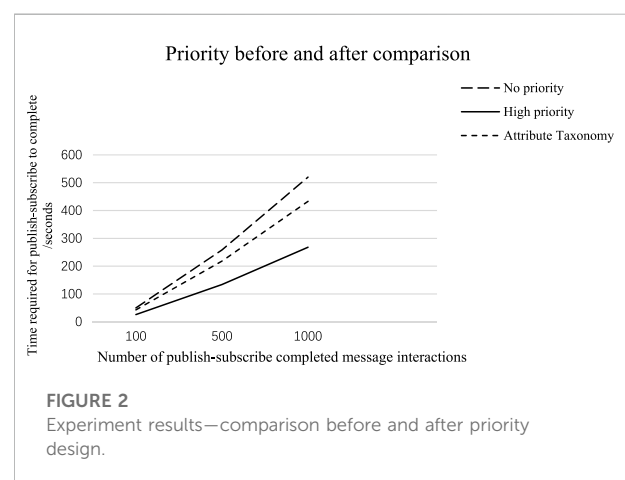
Experimental design

The system experiment is divided into the comparison of messages with or without priority, the comparison of feedback time periods, and the corresponding conclusions drawn through the comparison before and after the priority design. Comparing its data packet delay and data packet interaction time can know the performance of the priority design, and the feedback adjustment can get the impact of the time period established on the priority performance. At the same time, according to the comparison, the requirements for the stable operation of the system are also obtained. Finally,

the distribution frequency of the time period was tested, and the system's steady-state can be adjusted by changing the distribution frequency.

Experiment 1 Comparison of data before and after the priority of several subscribers when the port is blocked. Experiments were set up with multiple subscribers subscribing to their respective topic messages at the same time. The server distributes multiple numbers of subscription topics without setting topic ordering, and the distribution time increases as the number of distributions increases due to the limited distribution capacity of the port. Then, the server distributes multiple subscription topics in the set topic priority order. Observe the interaction time of higher priority topics in different experimental settings. Also, classification algorithms are involved in the comparison, taxonomy is a classification of subscription content with different attributes. In this paper, subscription content will be classified into several hierarchical classes based on their importance. The subscription classes with a high importance level are filtered out and can be distributed first.

It can be seen from the Figure 2 that the publish-subscribe priority designed in this paper can play a good role in the case of port congestion. The figure uses the comparison of data delay with and without priority to conclude that the design of priority can effectively reduce the delay time of data transmission in the case of port congestion. And it can be seen from the figure that as the amount of data increases, the delay time of high-priority data transmission and reception increases relatively smoothly, while in the case of no priority setting, the delay time increases by a multiple. From this, it is inferred that in the case of port congestion, the more data the better the role of the priority. In the industrial intelligent environment, it has a good auxiliary effect for the communication between the underlying devices and the upper-level data and can handle important and urgent



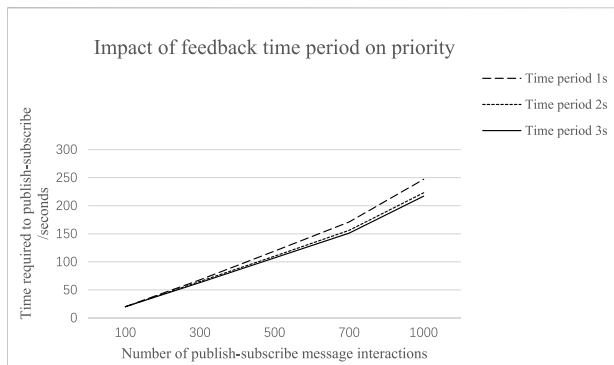


FIGURE 3 Experimental results ii effect of time period on priority.

messages. In addition, the topic prioritization designed in this paper allows for faster information interaction at higher priority levels and reduced packet loss at lower priority levels compared to traditional taxonomies.

Experiment 2 In the case of message congestion, high-priority messages get published and subscribed first and get transmission resources, which effectively reduces the transmission time and obtains information communication guarantee, but the transmission of low-priority messages will be affected. To prevent the situation that low priority cannot communicate with each other for a long time, feedback for low priority messages is designed. If the low priority message has not been transmitted after a certain period of time, the server will design the time period for the message to ensure the transmission of the message.

Figure 3 shows that the impact of different time periods on the priority is different. Publish and subscribe several times for low-priority messages in each time period. It can be seen from the figure that the longer the time period is designed, the more conducive to the communication of high-priority messages. However, the time period should not be too long to cause timeouts, and different time periods should be set up according to the needs of publishers and subscribers. Set up time periods that meet the requirements of low-priority subscribers without affecting the transmission of high-priority messages. Find an equilibrium point to ensure the equilibrium stability of the system.

Experiment 3 Low-priority subscribers request subscription packets from the server with different subscription frequencies. The server sets different time periods for low-priority messages. The message transmission delays for different distribution frequency states are derived in different time periods, and conclusions are drawn by comparing the delay ratios for different time periods.

The comparison of the Figure 4 and Figure 5 shows that the extension of the feedback time period brings the low priority

message delay ratio to a steady-state. At this time, the distribution frequency of low priority is fixed within the time period. By appropriately changing the server's distribution frequency of low-priority data, the transmission delay of low-priority messages can be reduced, and low-priority messages can be distributed faster. When the time delay ratio is 1, the state is the non-blocking state. The distribution frequency designed in this paper can make the delay ratio as close to 1 as possible, reducing the steady-state of low-priority delay and achieving faster low-priority message distribution. The design of the distribution frequency needs to comprehensively consider the impact of high and low priorities. Designing an appropriate distribution frequency can better serve priority messages. The distribution frequency of different requirements can be designed for subscribers with different requirements.

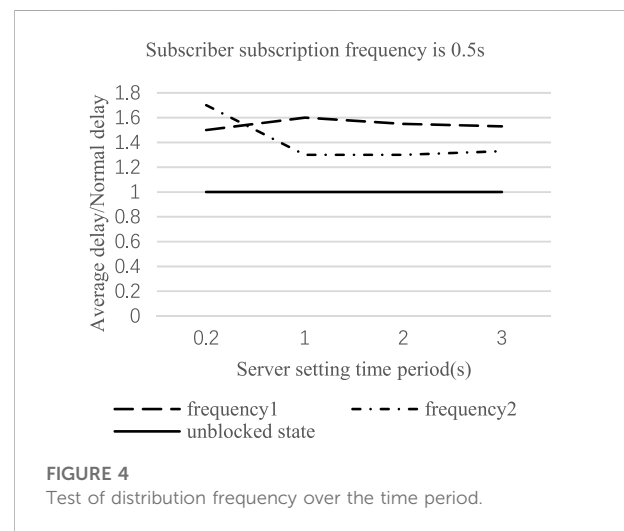


FIGURE 4 Test of distribution frequency over the time period.

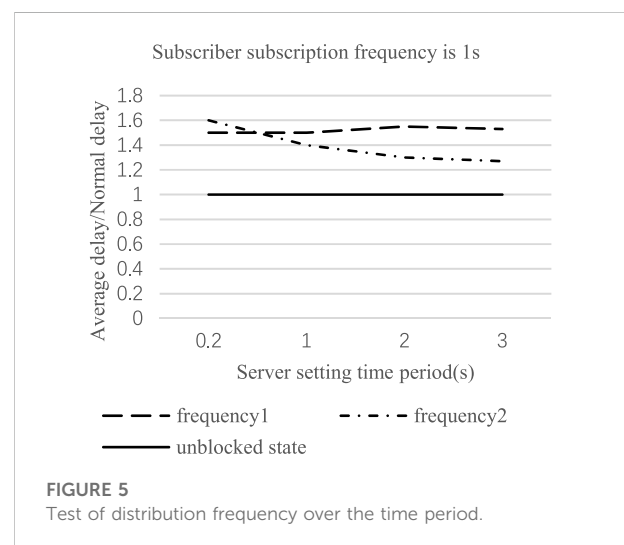


FIGURE 5 Test of distribution frequency over the time period.

Conclusion analysis

In the microgrid environment, for the situation that many underlying devices interact with the upper cloud data for information and generate message congestion. This paper completes the design of topic message priority for the publish-subscribe system, which can solve the problem of message congestion well in the case of port congestion. The experimental results also show that the message transmission delay can be reduced in actual situations, and the priority messages can be transmitted in a timely and stable manner. For some low-priority messages that cannot be transmitted for a long time, the server designed a personalized time period, subscriber requirements are different, and time periods are different. During the time period, the server can adjust the publish-subscribe frequency of low-priority messages to achieve faster message distribution. The next task is to further improve the feedback adjustment, and integrate and send data on the feedback priority, to achieve a faster distribution of messages.

Data availability statement

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

References

- Behnel, S., Fiege, L., and Muhl, G. (2006). "On quality-of-service and publish-subscribe," in 26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'06), Lisboa, Portugal, 4-7 July 2006, 20. doi:10.1109/ICDCSW.2006.77
- Cheng, B., Zhao, S., and Chen, J. (2018). "Remote monitoring and control web service for Internet of heating," in IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 23-27 July 2018, 785-786. doi:10.1109/COMPSAC.2018.00119
- Chi, X., Zhu, M., Yang, J., and Wang, W. (2020). "An MQTT-based scalable remote control Framework with NodeMCU and cloud platform," in 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), Harbin, China, 25-27 December 2020, 731-735. doi:10.1109/ICMCCE51767.2020.00161
- Ding, Yi, Fan, Binwen, Kong, Xiaoming, and Ma, Qianqian (2016). "Design and implementation of mobile health monitoring system based on MQTT protocol," in IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 03-05 October 2016, 1679-1682. doi:10.1109/IMCEC.2016.7867503
- Ericsson, G. N. (2010). Cyber security and power system communication—essential parts of a smart grid infrastructure. *IEEE Trans. Power Deliv.* 25 (3), 1501-1507. doi:10.1109/TPWRD.2010.2046654
- Fan, W., Liu, Y., and Tang, B. (2016). "Gem: An analytic geometrical approach to fast event matching for multi-dimensional content-based publish/subscribe services," in IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10-14 April 2016, 1-9. doi:10.1109/INFOCOM.2016.7524338
- Feng, S., Yin, W., and Wang, H. (2011). "Integrated micro-grid optimization and control technology," in International Conference on Advanced Power System

Author contributions

DC designed the project and completed experimental testing. DL conducted an in-depth review and guidance. RG provided thesis guidance.

Funding

This work was supported by the National Key R&D Program of China under Grant 2018YFE0205800.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Automation and Protection, Beijing, China, 16-20 October 2011, 2072-2075. doi:10.1109/APAP.2011.6180774

Feng, X., Jia, C., and Yang, J. (2008). "Semantic web-based publish-subscribe system," in IEEE International Conference on Service Operations and Logistics, and Informatics, Beijing, China, 12-15 October 2008, 1093-1096. doi:10.1109/SOLI.2008.4686562

Ganguly, B., and Chatterjee, A. (2020). "MQTT protocol based extensive smart motor control for electric vehicular application," in IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Prayagraj, India, 27-29 November 2020, 1-5. doi:10.1109/UPCON50219.2020.9376452

Hemayati, M. S., Mohsenzadeh, M., Seyyedi, M. A., and Yousefipour, A. (2010). "A Framework for fuzzy based semantic web service publish/subscribe," in IEEE Asia-Pacific Services Computing Conference, Hangzhou, China, 06-10 December 2010, 258-265. doi:10.1109/APSCC.2010.9

Huang, G., Yi, J., Zhao, R., Chen, J., Huang, M., and Wang, J. (2021). "Application of MQTT in power distribution Internet of Things," in 3rd International Conference on Electrical Engineering and Control Technologies (CEECT), Macau, Macao, 16-18 December 2021, 98-103. doi:10.1109/CEECT53198.2021.9672327

Kumar, A., and Johari, S. (2015). "Push notification as a business enhancement technique for e-commerce," in Third International Conference on Image Information Processing (ICIIP), Wanknaghat, India, 21-24 December 2015, 450-454. doi:10.1109/ICIIP.2015.7414815

Lee, S., Kim, H., Hong, D., and Ju, H. (2013). "Correlation analysis of MQTT loss and delay according to QoS level," in The International Conference on Information Networking 2013 (ICOIN), Bangkok, Thailand, 714-717. doi:10.1109/ICOIN.2013.6496715

- Lima, D. B. C., da Silva Lima, R. M. B., de Farias Medeiros, D., Pereira, R. I. S., de Souza, C. P., and Baiocchi, O. (2019). "A performance evaluation of raspberry pi zero W based gateway running MQTT broker for IoT," in IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 17-19 October 2019, 0076-0081. doi:10.1109/IEMCON.2019.8936206
- Ma, L., Cheng, S., and Shi, Y. (2021). Enhancing learning efficiency of brain storm optimization via orthogonal learning design. *IEEE Trans. Syst. Man. Cybern. Syst.* 51 (11), 6723-6742. doi:10.1109/TSMC.2020.2963943
- Ma, L., Huang, M., Yang, S., Wang, R., and Wang, X. (2021). An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization. *IEEE Trans. Cybern.* 52, 6684-6696. doi:10.1109/TCYB.2020.3041212
- Ma, Lianbo, Li, Nan, Guo, Yinan, Huang, Min, Yang, Shengxiang, Wang, Xingwei, et al. (2021). Learning to optimize: Reference vector reinforcement learning adaption to constrained many-objective optimization of industrial copper burdening system. *IEEE Trans. Cybern.*, 1-14. doi:10.1109/TCYB.2021.3086501
- Ma, Lianbo, Wang, Xueyi, Wang, Xingwei, Wang, Liang, Shi, Lin, Huang, Min, et al. (2021). Tcda: Truthful combinatorial double auctions for mobile edge computing in industrial Internet of Things. *IEEE Trans. Mob. Comput.*, 1. doi:10.1109/TMC.2021.3064314
- Merayo, N., Pavon-Marino, P., Aguado, J. C., Durán, R. J., Burrull, F., and Bueno-Delgado, V. (2017). Fair bandwidth allocation algorithm for PONS based on network utility maximization. *J. Opt. Commun. Netw.* 9 (1), 75-86. doi:10.1364/JOCN.9.000075
- MQTT (2020). The standard for IoT messaging[EB/OL]. Available at. <http://mqtt.org/>.
- Qian, J., Yin, J., and Dong, J. (2011). "Parallel matching algorithms of publish/subscribe system," in Eighth International Conference on Information Technology: New Generations, Las Vegas, NV, USA, 11-13 April 2011, 638-643. doi:10.1109/ITNG.2011.113
- Qiao, B., Jiang, N., Wang, Z., and Gao, F. (2015). "A new matching structure and interval division on content based publish/subscribe system," in IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on CyberSpace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems, New York, NY, USA, 24-26 August 2015, 641-646. doi:10.1109/HPCC-CSS-ICSS.2015.87
- Qin, Q., Chen, Z., and Wang, X. (2012). "Overview of micro-grid energy management system research status," in Power Engineering and Automation Conference, Wuhan, China, 18-20 September 2012, 1-4. doi:10.1109/PEAM.2012.6612478
- Sun, Z., Di, L., Fang, H., Zhang, C., Yu, E., Lin, L., et al. (2016). "Embedding Pub/Sub mechanism into OGC web services to augment agricultural crop monitoring," in Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics), Tianjin, China, 18-20 July 2016, 1-4. doi:10.1109/Agro-Geoinformatics.2016.7577653
- Wu, X., and Li, N. (2018). "Improvements of MQTT retain message storage mechanism," in 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 25-27 May 2018, 957-961. doi:10.1109/IMCEC.2018.8469192
- Xu, M., Lv, P., and Wang, H. (2014). "Match-Ladder: An efficient event matching algorithm in large-scale content-based publish/subscribe system," in Proceedings of the Winter Simulation Conference 2014, Savannah, GA, USA, 07-10 December 2014, 922-932. doi:10.1109/WSC.2014.7019952
- Xu, M., Lv, P., and Wang, H. (2013). "Predicate priority based event matching algorithm in publish/subscribe system," in 2013 Fourth International Conference on Networking and Distributed Computing, Los Angeles, CA, USA, 21-24 December 2013, 146-150. doi:10.1109/ICNDC.2013.38
- Yan, B., Gao, C., and Guo, X. (2010). "A new marginal revenue based scheduling algorithm for publish/subscribe systems," in 7th International Conference on Service Systems and Service Management, Tokyo, Japan, 28-30 June 2010, 1-5. doi:10.1109/ICSSM.2010.5530261
- Yan, W., Muthusamy, V., Chen, M., Huang, Y., Xiao, Z., Pei, X., et al. (2013). "Queue reorganization for subscription congestion avoidance in publish/subscribe systems," in IEEE 32nd International Performance Computing and Communications Conference (IPCCC), San Diego, CA, USA, 06-08 December 2013, 1-10. doi:10.1109/IPCCC.2013.6742765
- Yang, J., Fan, J., and Jiang, S. (2016). "Doco: An efficient event matching algorithm in content-based publish/subscribe systems," in IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS), Wuhan, China, 13-16 December 2016, 200-207. doi:10.1109/ICPADS.2016.0035
- Yang, K., Zhang, B., Zhang, J., and Zhu, J. (2021). "Design of remote control inverter based on MQTT communication protocol," in IEEE International Conference on Mechatronics and Automation (ICMA), Takamatsu, Japan, 08-11 August 2021, 1374-1378. doi:10.1109/ICMA52036.2021.9512665
- Ying-Ying, Y. E. (2014). Personalized priority routing algorithm in publish/subscribe network. *Journal of Chinese Computer Systems* 35.2, 199-209.
- Yu, H., Luan, W., Sheng, W., Liu, H., and Ma, W. (2015). "Evaluation of the standardization needs of microgrids technology," in 5th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT), Changsha, China, 26-29 November 2015, pp. 2144-2148. doi:10.1109/DRPT.2015.7432604
- Yue, M., Ruiyang, Y., Jianwei, S., and Kaifeng, Y. (2017). "A MQTT protocol message push server based on RocketMQ," in 10th International Conference on Intelligent Computation Technology and Automation (ICICTA), Changsha, China, 09-10 October 2017, 295-298. doi:10.1109/ICICTA.2017.72
- Zhai, L., Sun, L., and Liu, Y. (2008). "Modeling and evaluation of high-performance publish-subscribe system," in International Symposium on Computational Intelligence and Design, Wuhan, China, 17-18 October 2008, 457-460. doi:10.1109/ISCID.2008.47
- Zhang, F., Yang, Y., Yang, Y., and Jin, B. (2015). "Optimizing the quality of service for a publish/subscribe system," in IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), Beijing, China, 10-14 August 2015, 424-431. doi:10.1109/UIC-ATC-ScalCom-CBDCCom-IoP.2015.92