Applied Mechanics and Materials *Online: 2014-05-23*
ISSN: 1662-7482, Vols. 556-562, pp 5321-5327
doi:10.4028/www.scientific.net/AMM.556-562.5321

The Study on the Methodology of Transactionevent in the Ale of Rfid Networks

Zhao Huiqun [1] ,Yang Haigang[1]

(Institute of Information Engineering, North China University of Technology, Beijing100144, China)

Keywords: Internet of Things, ALE middleware, ALE, TransactionEvent

Abstract. TransactionEvent is one of the five events defined in EPCGlobal standard. As TransactionEvent lasts for a long period and processes large data, it has a higher demand of real-time. The process of the TransactionEvent in the Internet of Things is complex. In order to overcome these disadvantages, this paper proposes a non-integrated program. This program will ensure the TransactionEvent processing efficiency, reliability and real time. In the end of this paper, the article will implement a prototype system of a commercial IoT to verify this method.

Introduction

In accordance with the RFID IoT technology reference standard given by the International Organization for Standardization of EPCGlobal, RFID networking provides a set of network services, including Reader Management(RM)[1], EPC Information Service(EPCIS)[2],Application Layer Event(ALE)[3] and Object Naming Service(ONS)[4].ALE layer locates between the EPC event detection layer(RFID tag reading level) and commercial application logic layer(EPCIS layer).ALE is the nerve center of IoT technology. It is also the intermediary between sensing layer and applications[5].

In the EPCGlobal ALE specification, it defines a number of interface functions, but it didn't illustrate a specific implementation. The main purpose is to provide a full-featured alternative middleware model. Middleware platform will has good resistance, scalability, convergence and versatility if it is compliant to the specification[6]. ALE in the EPCGlobal network architecture major tasks are as follows.

1. Providing a set of communicate services with hardware devices, shielding the difference between different hardware, and capturing EPC tags from various devices.

2. Providing data processing and managing services, integrating and filtering data that repeating and non-interested, reducing the amount of data and re-group counts.

3. Providing communication service with application. According to the needs of enterprise, it packages the processed data into different formats and sends reposts to enterprise applications.

EPCIS defines five types of events to meet the business needs of enterprise. These five event types are EPCISEvent, Objectevent, AggregationEvent, QuantityEvent, and TransactionEvent. So ALE must provide interface and service for the events, this paper mainly studies on the TransactionEvent in ALE layer.

TransactionEvent describes the association or disassociation of physical objects to one or more business transactions[7]. Due to the nature of the transaction, it results three aspects. 1. sBecause we can't confirm the time of the sale, the transaction processing cycle of TransactionEvent is uncertainly and very long mostly. 2. Transaction Projects are often large, resulting in a large transaction data and dirty. 3. TransactionEvent demands real-time process. Traditional centralized processing method exists insufficient in the aspects of efficiency; reliability and real-time. To solving these problems, this paper summarizes a new approach, that is, non-centralized control,

gives a solution and the main algorithm, and tests practical application of the above solution combining with a commercial enterprise of IoT.

For convenience, this paper is divided into four parts. Part I will briefly introduce information exchange and TransactionEvent in the Internet of things. Part II will specifies the solutions and algorithmic belong to the non-centralized control method. Part III will present an implement of the solution. Part IV will do a experiment and evaluate the results. Finally, the paper will introduce related work and conclude the research work.

IoT TransactionEvent

In order to solve the problem in the IOT TransactionEvent, this part briefly introduces some information about the message exchange and TransactionEvent.

Information Interaction of IoT. Base on ALE specification, how the message is exchange between the reader and middleware and application is show in the figure below.

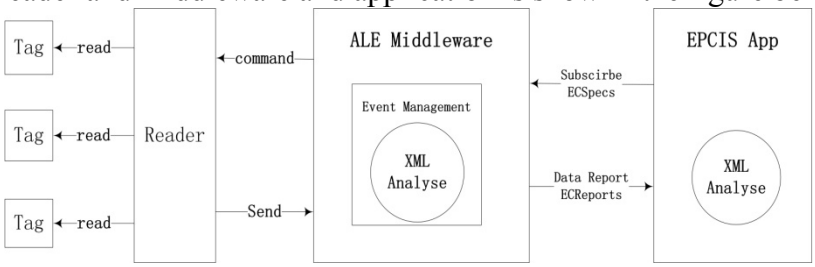

Fig1. Message exchange in RFID IoT

EPC ALE protocol is the interface protocol between EPC middleware reader module and client application. The protocol defines how customers to filter and integration EPC tag from multiple readers and defines a unified interface facing different enterprise applications. Thu even if the back-end application increases, is replaced by other software, or RFID type changes, the other parts needn't to be modified[8].

TransactionEvent. In the EPCglobal standard, only EPC Information Service (EPCIS) Version 1.0.1 Specification provides TransactionEvent data model, but do not have the specific implementation details of the deal. This is the model given by EPCIS.

Table1. TransactionEvent data model in EPCIS

TransactionEvent
eventTime:Time
recordTime:TIme
eventTimeZoneOffset:String
paraentID:URI
epcList:List<EPC>
readTime:Time
bizStep:BiZStepID
Disposition:DispositionID
readPoint:ReadPointID
bizLocation:BizLocationID

Property contains eventTime, recordTime. In addition, there are another four important characteristics.

"What"，physics or objects, who participate in the event；

"When"，date and time, when event occurrence；

"Where"，place, where occurrence an event；

"Why"，reason, related to business environment,；

The basic data is represented by these types of vocabulary, that is, Readpoint, BusinessLocation, Disposition, BusinessTransaction , BusinessStep, EPC list[9].

The communication between middleware and application is charged by Event Management Module. Its essence is that the application definition file EPCsec includes TransactionEvent required data, Event Management Module analyzes file ECSec, calls the Readers to read data, processes (filtering, grouping, aggregation) data according to the demand, then integrates into a ECReport in the Event Management Module, and returns the ECReport to the application in the format of XML[10].

Methodology

The essence of TransactionEvent is query the trading information of the items belong to the trading project. Traditional TransactionEvent processing method in IoT middleware is centralized control mode, which means only when all the EPC tag related to the event has been collected completely, or we can see that all items belong to the trade project sales, ALE will create report and send it to EPCIS and then finish the event.

This paper presents a new method to process TransactionEvent. The main idea of the scheme is dividing a big and long duration TransactionEvent into several small time slices and creating a fragment report during every single time slice. Reports' sending is separated from generating. All the fragment reports will be sent to EPCIS, assembled by EPCIS, and then become the final TransactionReport. We only discuss the processing method in ALE middleware.

Solution analysis. The solution is divided into two parts.

(1) Generating Reports. No-centralized control method needs to divide a whole long time into a number of time slices, then you need to set up a TIME thresholds. This TIME thresholds will fix the duration of each time slice. Generating fragment report is intended to reduce the size of the report. However it will waste time if a fragment report is too small. So you should set a COUNT threshold too, and this COUNT threshold will stipulate how many tags a fragment should include at least. The generation process and steps of creating fragment report are as follows.

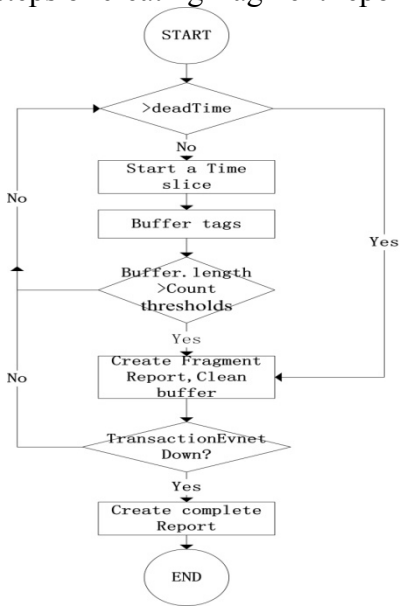

Fig2 The flow chart of the data report generation

(2)Sending Reports. After generating a fragment report, you should put the name of the report and sequence number into a sequence list. It will be helpful when sending reports. Now, fragment reports should be send to upper application layer. To ensure the reports sent safety, here we will use a method similar to TCP/IP, of which sending and responding mechanism are based on connection.

1. Connect with the EPCIS port.
2. Send the generated reports to EPCIS, according to the sequence list.
3. EPCIS sends confirm message to ALE after receiving fragment reports from ALE.
4. ALE receives an acknowledgment message of the client, and then deletes the number of the confirmed fragment report from the server sequence list. If ALE did not receive confirmation message of the client, the server will send the unacknowledged fragment report at the next sending period.

Algorithm. The first is generating fragment report. The server processes every EPC tag read at every time slice. The algorithm is as follows.

```
private Report create(List<EPC > buffer){
List<Record> records=null;
for(EPC epc:buffer){
Record record = new Record(epc);
records.put(record);
}
Document document = null;
for(Record record: records){
put record into document;
}
return document;
}
```

Fragment reports are sent to EPCIS by the ALE after generated. Sending thread and receiving thread is obviously two threads, and the both threads share the sequence list. The algorithm is as follows.

```
void    run(){  //send report
lock fragementReportList;
if the fragementReportList is not null;{
sendReport base on the fragementReportList;
}
unlock fragementReportList;
}
void run(){          //get confirm message from EPCIS
get confirmMessage;
lock fragementReportList;
if(confirmReport in fragementReportList){
remove confirmReport from fragementReportList;
}
unlock fragementReportList;
}
```

The Implementation of the Methodology

According to the description of the main solutions above part, as well as the EPGGlobal standard, here give an implementation of the scheme.

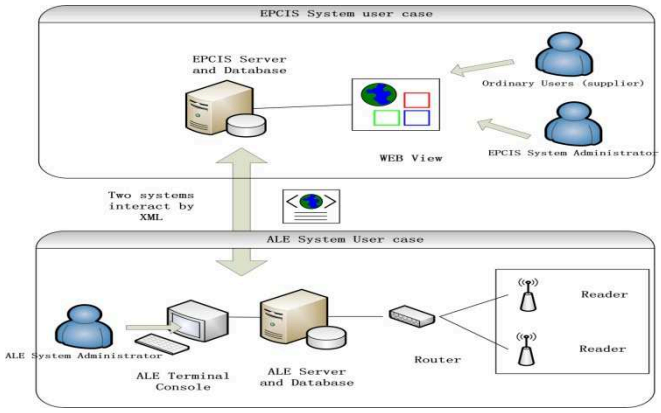

Fig3 IoT prototype System user case

1. The supplier lands EPCIS system, and fills TransactionEvent subscription request file. Then the file is formatted to XML and sent to ALE by EPCIS server;

2. ALE layer receives the subscription file, monitors the readers, generates data report in the form of XML based on the data meeting the conditions, and sends the XML file to EPCIS system.

3. EPCIS combines fragment data reports into the final TransactionReport , and provides the TransactionReport for user to view。

Data Structure. In the comminuting of TransactionEvent between EPCIS and ALE, the main class file information is TransactionEvent subscription file and data report file.

Table2 TransactoinSubscribe Model and FragmentReport Model in system

TransactionSubscribe	TransactionFragmentReport
Id: int	Id: int
subID:String	subID: String
endTime: Date	fromTime: Date
epcList: List<EPC>	endTime: Date
recordTime: boolean	reportCount: int
readPoint: boolean	reportStartNum: int
bisposition: boolean	isDown: boolean
	isDownNormal: boolean
	epcRecords:EPCRecord

TransactionEvent subscribe file is created by EPCIS and sent to ALE from EPCIS. The contents are specified by EPCGlobal protocol which has been presented before. Fragment report format shall be formulated by both middleware and application. TransactionEvent fragment data report file is generated by ALE layer, and sent to the EPCIS layer by ALE. The reports are feedback on the subscription request, and contain trading information of the items that are read in time slice. All of the fragment report files will be integrated by EPCIS, and get the final TransactionEvent report.

Input and Output: Input: A TransactionEvent subscribe. Output**:** A fragment report. This report is one of the reports created in the TransactionEvent period, and it contains detail information of the goods that traded in the time slice between 15:37 and 16:37.

```
<?xml version="1.0" encoding="gbk" standalone="yes"?>
<transactionSubscribe>
  <subID>130926151047</subID>
  <eventTime>2013-09-26T15:10:47.765+08:00</eventTime>
  <endTime>2013-09-26</endTime>
  <eventTimeZoneOffset>+08:00</eventTimeZoneOffset>
  <recordTime>0</recordTime>
  <bizStep>1</bizStep>
  <disposition>0</disposition>
  <readPoint>0</readPoint>
  <bizLocation>1</bizLocation>
  <bizTransaction>0</bizTransaction>
  <epcList>
    <epc>urn:epc:id:sgtin:0888892.000008.118</epc>
    <epc>urn:epc:id:sgtin:4193364.005116.71</epc>
    <epc>urn:epc:id:sgtin:4193364.005116.57 </epc>
    <epc>urn:epc:id:sgtin:0015380.005116.255</epc>
    <epc>urn:epc:id:sgtin:0858172.000008.255</epc>
    <epc>urn:epc:id:sgtin:1333332.005116.248</epc>
    <epc>urn:epc:id:sgtin:1333332.005116.88</epc>
    <epc>urn:epc:id:sgtin:0000444.000088.101</epc>
    <epc>urn:epc:id:sgtin:0000444.000008.255 </epc>
    ......
  </epcList>
  </transactionSubscribe>
```

Fig4. An Input Example

```
<transactionreport>
  <subID>130927163603</subID>
  <startTime>2013-09-27 15:37:20</startTime>
  <endTime>2013-09-27 16:37:20</endTime>
  <recordtime>2013-09-27 16:37:24</recordtime>
  <down>0</down>
  <normal>0</normal>
  <sequenceNumber>1</sequenceNumber>
  <records>
```

Fig5. An Output Example

System Testing and Evaluation

After system implementation, the paper analyzed and evaluated the performance of the system through an experiment. The experiments evaluated mainly from two expects, the memory utilization and the loss of report (reliability).

The system simulates a situation that receives TransactionEvent item from the EPCIS layer, including 1400 articles, trading 200 items daily, and lasting seven days. Meanwhile, the system will record the memory utilization and the sending and loss of the report.

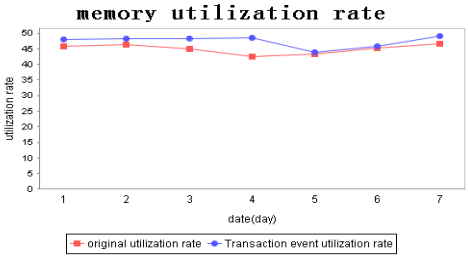

Fig.6 Memory utilization rate

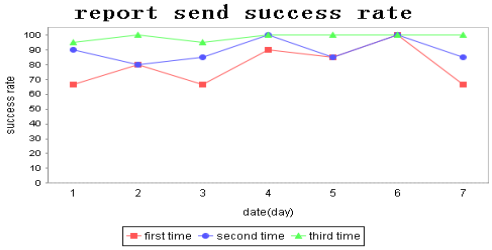

Fig.7 Reports send success rate

Original utilization is measured when the server does not process any events, and the utilization of trade events is measured when the server processes trade event only. As can be seen from the chart, in the cycle of the TransactionEvent, memory utilization is usually slightly higher than the usual memory utilization, thus greatly reducing the load on the server, and separating more resources to deal with other matters.

The success rate of first transmission is not high, and the reason for this phenomenon may relate with the network congestion or the busy of EPCIS port. However, no matter what the reason, in the case of the fragment reports failed to be sent at the first time, the internal mechanism will send the failed files in a certain time interval, until you get an acknowledgment message. This will ensure the best success rate of sending reports, thus ensuring the reliability of event processing.

Related Works and Conclusion

Research for RFID IoT event processing method has been a hot issue. Xi'an University of Technology proposes CEP (Complex Event Processing) technology to be used in the Internet of Things[11]. CEP technology primarily gets applied in Event Driven Architecture (EDA) system. It is helpful to realize many business applications. It put forward s general structure model of CEP system that consists 9 modules. In event model, CEP is only able to perform matching fixed event model, and can not to perform dynamic event model. Difficult of CEP research focuses on processing mode, processing capability and security mechanism. Hu'nan University proposes a distributed Context-Aware Complex Event Processing Method[12]. Event Processing is the most important part of the middleware of IoT. The method is based on the representation and reasoning of fuzzy ontology of event context. Query rewriting supports event context processing. Distributed processing and heuristic optimization is supported by query planning and data classification. This method can process fuzzy context and scalability of this approach. It is better than normal method when processing big data event in IoT.

Using non-integrated approach to deal with the TransactionEvent in the Internet of Things can improve the processing efficiency of TransactionEvent by cutting the report into pieces. It reduces the existing time of the electronic tag in the memory space, and thereby improves the memory utilization. It also improves the reliability by using the method similar with TCP / IP packet delivery mechanism. The more a big deal for the project, the greater the advantage of this method is shown.

Natural Science Foundation of China (Grant No:61070030,61370051). Beijing Government and Education Committee (Grant No:PHR201107107) 1.Huiqun Zhao, Professor/Dr, research areas:IoT technologies. Address:Institute of Information Engineering, North China University of Technology, Beijing100144,China. Email:zhaohq6625@sina.com 2.Haigang Yang, MA, research areas: IoT middleware. Address:Institute of Information Engineering, North China University of Technology, Beijing100144,China. Email:yanghaigang11@163.com.

References

[1] EPCglobal, "Reader Management 1.0.1," EPCglobal Ratified Standard, May 2007, http://www.epcglobalinc.org/standards/ rm/rm_1_0_1-standard -20070531.pdf.

[2,10] EPCglobal, "The Application Level Events (ALE) Specification, Version 1.0," EPCglobal Ratified Standard, September 2005, http://www. Epcglobalinc.org /standards/ ale/ale_1_0 -standard -20050915.pdf.

[3,7,9] EPCglobal Ratified Standard, October 2005, http://www.epcglobalinc.org/ EPCglobal,EPC Information Services(EPCIS)Version1.0 Specification, EPCglobal Ratified,Standard,April2007,http://www.epcglobalinc.org/standards/epcis/epcis_1_0-standard-20070412.pdf.

[4] EPCglobal, EPCglobal Object Naming Service (ONS), Version 1.0," EPCglobal Ratified Standard, October 2005, http://www.epcglobalinc.org/ standards /ons/ons _1_0- standard-20051004.pdf

[5] Zhang Xiaopeng, LI Jianmin, Lin Zhenrong. Complex Event Handling Technology in the REFID Middleware[J]. Micro Computer Information(Embedded and SOC), 2009,25(9-2):145.

[6] Guo Yuehui , Prof.Y.Liu Research and design of RFID middleware in the Internet of things. May 2011.

[8]Lu Kang,Zhang Qifeng, Design and Implementation of RFID Middleware Base on JMS.TP311 1003-4978（2008）05-0537-04.

[11] JING Xin, ZHANG Jing, LI Junhuai. Overview of complex event processing technology and its application in logistics Internet of Things. Computer Applications. 1001-9081(2013)07-2026-05.

[12] Cao Kening, Wang Yongheng, Li Renfa, Wang Fengjuan. A Distributed Context- Aware Complex Event Processing Method for Internet of Things. Computer Research and Development. 2013,50(6).