A Rapid Response Approach Applying Edge Computing for Distributed Warehouses in WSN

Yuechun Wang[1], Ka Lok Man[2], Danny Hughes[3], Steven Guan[1], Prudence Wong[4]

[1] Department of Computer Science and Software Engineering (CSSE), Xi'an Jiaotong-Liverpool University(XJTLU), Suzhou, China. {yuechun.wang, steven.guan}@xjtlu.edu.cn

[2]AI University Research Centre (AI-URC) and Department of Computer Science and Software Engineering (CSSE), Xi'an Jiaotong-Liverpool University (XJTLU), Suzhou, China; Swinburne University of Technology Sarawak, Malaysia. ka.man@xjtlu.edu.cn

[3] imec-DistriNet, KU Leuven, Leuven, B-3001, Belgium. danny.hughes@cs.kuleuven.be

[4]Department of Computer Science, University of Liverpool, UK. p.wong@liverpool.ac.uk

Abstract. This paper presents a rapid response system architecture for the distributed management of warehouses in logistics by applying the concept of edge computing. A tiered edge node architecture is proposed for the system to process computing tasks of different complexity, and a corresponding rapid response algorithm is introduced. A software-defined simulation is done to evaluate the system performances on rapidness and correctness, from which it can be concluded that all pre-defined emergency cases can be detected and responded to within a relatively short period of time.

Keywords: edge computing, rapid response algorithm, Wireless Sensor Networks (WSN), distributed warehouses management, logistics.

1 Introduction

As a mature Internet of Things (IoT) scenario, Wireless Sensor Networks (WSN) have rapidly proliferated over the last decade. These diversified WSN applications are emerging rapidly, while the quantity of various nodes and platforms in the WSN is increasing exponentially. Wearable human sensor network, the smart home, intelligent logistics and transportation [1], as well as the smart city [2] are typical application scenarios for current WSNs.

From the perspective of logistics, the warehouse is a critical scenario for contemporary intelligent logistics applications. For warehouse management, sensed data may be used for two general purposes: one is for cargo management, which includes goods identification (using RFID) and goods tracking (location and movement); the other is for safety management, which refers to the environmental monitoring and data security. The logistics companies who aim to conduct business nationwide need to consider both the centralized global control of the profession on Cloud as well as the management of distributed networks of warehouses locally. At this scale, the traditional WSN plus Cloud mode may lead to either high bandwidth use or latency in undertaking emergency interventions. In short, many applications require both WSN localization and Cloud globalization which cannot be satisfied by a simple WSN-Cloud architecture. In such case, a well-designed WSN-Edge-Cloud system architecture that integrates edge computing features with the WSN-Cloud architecture would solve these problems and improve the efficiency of the business [3, 4].

Being regarded as a relay between the data centre on the Cloud and sensor nodes in the WSN, edge computing nodes extend the Cloud Computing paradigm to the edge of the network in a bidirectional way. On the node-to-cloud direction, edge nodes revolve around local functionality for geographically closer sensing area with the feature of data pre-processing and rapid reaction [5]. These outcomes will be sent to the Cloud selectively, according to the explicit application requirements. In the cloud-to-node direction, edge nodes achieve distributed deployment of the broad class of applications under the macro control of the Cloud and perform the tasks allocated by the Cloud [6].

2 Architecture and Methodology

2.1 Edge Computing-based Graded System Architecture

The edge computing-based graded system architecture, which can be differentiated from traditional WSN system architecture, consists of three general layers at the vertical direction, which from top to bottom are the Cloud, Edge, and WSN infrastructure as shown in Figure 1. As a widely accepted environmental sensing infrastructure, sensor nodes in the WSN collect sensing data and track changes of the environment continuously. For better identification and management, sensor nodes in the WSN are logically separated into different areas.

The Edge computing layer is introduced into the system by considering it as the implementation of cloud computing close to the physical environment. The

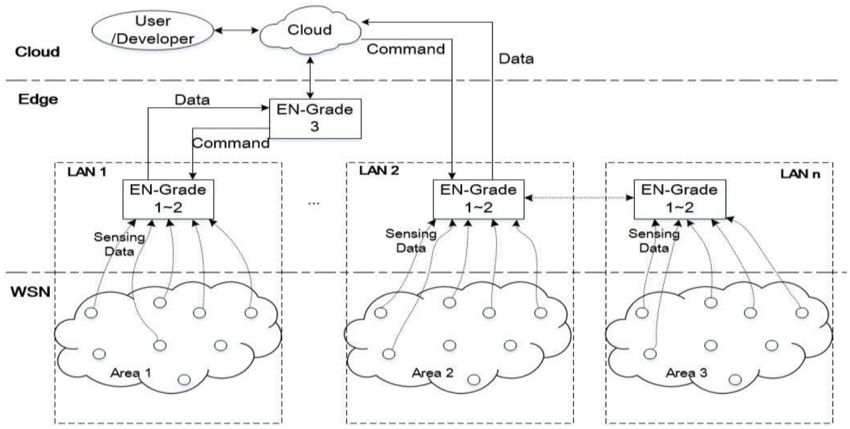


Figure 1. Edge computing-based tiered system architecture. ‘EN’ indicates edge node. A smaller EN-Grade number indicates the physically closer location from the edge node to the sensor nodes and sensing devices.

functionality of edge computing is refined into three grades of edge nodes. Grade one and two edge nodes are focused on the data formatting, preliminary data processing for WSN data collection, as well as the execution of tasks and control commands allocated by the upper layer (higher grade edge nodes or the cloud). Grade three edge nodes contribute to more complex data analysis, which involves data that is potentially useful for prediction and control, as well as generating or relaying control commands from the upper layer to the lower layer.

The cloud layer contributes to the centralized analysis of global data and management of the entire network. In addition, the connection between users and the system via the cloud realizes the remote operation and control all areas covered by the terminal devices. For application developers, the system can be accessed via the cloud or edge node for application deployment regarding the deployment requirements and the network condition.

2.2 Rapid Response Algorithm

Within a target monitoring area, there are two primary cases in which sensor nodes may generate abnormal sensing data: one is the sudden environmental change, the other is the error data caused by sensor broken or irruption. A rapid response is only

Algorithm 1. Rapid Response Algorithm

```
Inputs:
  Bound threshold: TH_b
  Observe period 1: T_1
  Data set: D(n)={d(n,i),n∈N,i:timestamp}
  Data status: C(d)={true,when d≥TH_b;false,when d<TH_b,d∈D}
  Node set: N={n_1,n_2,…,n_k,k:number of nodes in the area}
  Neighbor node set: N_nb(n)
  Critical percentage: p
  Observe period 2: T_c
  Data gradient set: Grad(n)={grad(n,i),n∈N,i:timestamp}
  Data gradient: grad(n,i)=[d(n,i+T_c)-d(n,i)]/T_c,n∈N,  i:timestamp
Process on EN-1
  if d(n,i)>TH_b:
     observe C(d), where d∈D_nb,D_nb={d(m,j),m∈N_nb(n),j∈(i,i+T_1)}
     if C(d)=true:
        activate Alarm-1
     else :
        generate M=(n,i)
        activate EN-2
Process on EN-2
  observe M
  compute Grad(m), where m∈N_nb(n)
  if #{grad(m,i)>0}/#{N_nb}>p:
      activate Alarm-2
  else :
      if grad(n,i>0):
          activate Alarm-3
      else :
```

expected to be triggered by the first case, which could save time for emergency interventions and reduce the potential for business losses. In contrast, a rapid response caused by the second case will lead to a waste of resources.

The rapid response algorithm proposed in this paper as shown in Algorithm 1 is under the premise of ensuring accuracy, which classifies the urgent cases into three types: Rapid Growth, Slow Growth-Diffusion, and Slow Growth-Non-Diffusion.

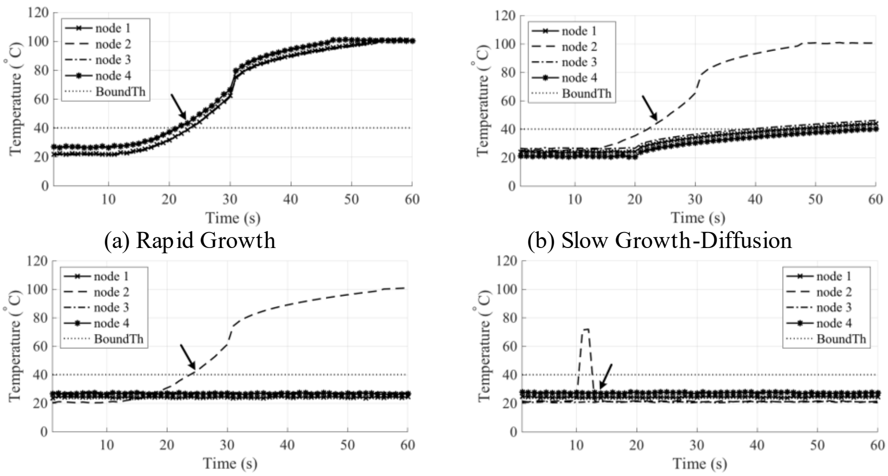
Threshold setting is one of the most popular approaches to distinguish abnormal data and normal data. At the Grade-1 edge nodes, the bound threshold (TH_b) is set for real-time comparison. Any sensing data collected by a sensor node that upload to the Grade-1 edge node will be compared with TH_b. There are two cases that may happen at the Grade-1 edge node by comparing real-time sensing data with TH_b: a) more than one sensor nodes are distinguished as abnormal within a short period (say T_1), b) abnormal data appear on a single node. For case a), we consider a Rapid Growth case happened and generate Alarm-1 directly; while for case b), Grade-1 edge node will trigger Grade-2 edge node with a new generated message which includes abnormal sensor node ID for further computing and judgement. Once Grade-2 edge node be triggered by this message, it starts to analyze the trend of sensing data of both the abnormal node and its neighbor nodes. The trend is measured by computing the gradient of data in adjacent time point. The percentage of neighbor nodes who have the same trend as abnormal one will decide the urgent case type: if there are more than p (a given percentage) neighbor nodes have the same trend, we consider a Slow Growth-Diffusion case happened and generate Alarm-2. Otherwise, the trend of the abnormal node in the coming period of time (say T_c) decides the urgent case type. A continuous change of sensing data will denote a Slow Growth-Non-Diffusion case happened and an Alarm-3 will be generated; while the stable sensing data indicates that an error has occurred and there will be no alarm message generated.

3 Implementation

In our simulations, there are four cases considered as listed in Table 1. Case 1-3 are corresponding to the three urgent cases types as introduced in Section 2.2, which are Rapid Growth, Slow Growth-Diffusion, and Slow Growth-Non-Diffusion. Case 4 indicates unexpected error data occurs on a single node. Each of the cases corresponds to an alarm type, which is generated by the edge node to distinguish the cases.

Table 1. List of all cases for experiments

Cases	Specification	Alarm type
case 1	Abnormal of environment is observed by a group of nodes (Rapid Growth)	Alarm-1
case 2	Abnormal of environment is observed by single node, the abnormal is diffusion (Slow Growth-Diffusion)	Alarm-2
case 3	Abnormal of environment is observed by single node, the abnormal is non-diffusion (Slow Growth-Non-Diffusion)	Alarm-3
case 4	Error data on single node	None



(a) Rapid Growth (b) Slow Growth-Diffusion

(c) Slow Growth-Non-Diffusion (d) Error data

Figure 2. Simulation on correctness and rapidness performance of the algorithm

Corresponding to three real-world scenarios that produced temperature sensing data, which are open flame spread, high-temperature steam leakage and diffusion, as well as node device over temperature, we generate three sets of data by software for the experiments on case 1-3. The error data for case 4 is inserted into a data set simulated the indoor environmental change manually. All the simulated sensing data are sampled once per second during the experiments.

The performances on correctness and rapidness corresponding to sub-figures (a)-(d) in Figure 2 are tested under four simulated environments. For each test, there are four nodes updated sensing data over 60 seconds to the edge node simultaneously. Referring to the input parameters listed in Algorithm 1, the 'bound threshold (TH_b)' is set to be 40 while the 'observe period one (T_1)' is set to be 1. The 'observe period two (T_c)' is 5 and the 'critical percentage (p)' is 0.5 in our experiments.

The system response time is pointed out by an arrow on each sub-figure, which corresponding to the timestamp 23 sec, 24 sec, 25 sec, and 13 sec. Comparing with the bound threshold line (labelled as *BoundTh* in the figure), it can be observed that all the emergency cases are detected and responded within an 'observe period two'.

4 Conclusion

A rapid response system architecture is proposed in this paper, which involves the concept of edge computing in WSN. From the perspective of distributed warehouse management in logistics, an algorithm for distinguishing and rapidly responding to emergency cases is introduced. Tested by a software-defined simulation, the performance on the correctness and rapidness of the Grade-1 and Grade-2 edge nodes in the system applying the rapid response algorithm shows that all pre-defined emergency cases can be detected and responded within a relatively short period of time.

To implement the entire system architecture as proposed in this paper, a clear direction for future research is the implementation of Grade-3 edge nodes, which potentially focuses on the short-time prediction. Besides the edge computing layer, the interaction and interoperation between the edge and the Cloud is also a valuable direction to extend our research.

Acknowledgement

This project is supported by the Research Development Fund (#RDF14-03-12) and XJTLU Key Programme Special Fund (#KSF-P-02) of the Xi'an Jiaotong-Liverpool University, Suzhou, China.

References

1. Xiao, X., et al., Developing an intelligent traceability system for aquatic products in cold chain logistics integrated WSN with SPC. Journal of Food Processing and Preservation, 2016. 40(6): p. 1448-1458.
2. Mohamed, N., et al., SmartCityWare: a service-oriented middleware for cloud and fog enabled smart city services. Ieee Access, 2017. 5: p. 17576-17588.
3. Bonomi, F., et al. Fog computing and its role in the internet of things. in Proceedings of the first edition of the MCC workshop on Mobile cloud computing. 2012. ACM.
4. Mihai, V., et al. Wireless Sensor Network Architecture Based on Fog Computing. in 2018 5th International Conference on Control, Decision and Information Technologies (CoDIT). 2018. IEEE.
5. Zhang, G. and R. Li, Fog computing architecture-based data acquisition for WSN applications. China Communications, 2017. 14(11): p. 69-81.
6. Carrega, A., et al., A middleware for mobile edge computing. IEEE Cloud Computing, 2017. 4(4): p. 26-37.