

Proposal of a Distributed Cooperative IoT System for Flood Disaster Prevention and Its Field Trial Evaluation

Shiji Kitagami^{1,*}, Vu Truong Thanh¹, Dang Hoai Bac², Yoshiyori Urano¹,
Yohtarō Miyanishi³, Norio Shiratori¹

¹Global Information and Telecommunication Institute, Waseda University, Tokyo, Japan

²Posts and Telecommunications Institute of Technology, Hanoi, Vietnam

³ISEM, Inc. (Information Systems Engineering and Management), Tokyo, Japan

Abstract Effective applications of IoT (Internet of Things) for disaster prevention are highly expected. In particular, in applications for flood disaster prevention, an early warning of the flash flood caused by locally heavy rain is required as well as a flood impact analysis based on water level and rainfall monitoring in the whole drainage basin. However, in a conventional server-centric IoT system, an increase of network load and a delay of feedback control has become one of the most critical issues. Therefore, it is difficult to apply the IoT system to applications of flood disaster prevention, in which both a real-time flood alert and a flood impact analysis are required. In this paper, we propose a distributed cooperative IoT system to solve such problems, and make an attempt to apply the proposed system to applications for flood disaster prevention. This proposed system provides an event-driven data collection mechanism for efficient data collection in low-bandwidth networks on the Never Die Network to achieve disaster tolerance of the access network. In addition, a rule-based autonomous control mechanism as the edge computing in the IoT gateway for a real-time flood alert is offered. In order to evaluate the effectiveness of our proposed method, we developed a prototype system and conducted a field trial in Quang Nam Province, Vietnam. As the result of our evaluation, we have confirmed that up to 1/20, the proposed method can reduce the network load for flood monitoring, and can issue the flood warning at proper timing.

Keywords Server-centric IoT System, Edge Computing, Intelligent IoT Gateway, IoT Coordinate Server, Never Die Network, Event-driven Data Collection, Rule-based Autonomous Control

1. Introduction

Recently IoT systems, in which many sensors or devices are connected directly to the internet to provide various services without human intervention, have been attracting attention [1]. Applications of IoT system are adopted in the industrial sector, the household sector and the social sector. In the industrial sector, the increasing sophistication of remote maintenance and supply chain management is expected [2]. In the home sector, the increasing sophistication of health care and energy management is expected [3, 4].

In the social sector, effective applications of the IoT system for disaster prevention, such as monitoring and alarm for a flood, are anticipated [5]. In applications for flood disaster prevention, it is necessary to issue an alarm of flash floods caused by locally heavy rain. In addition, it is crucially important to conduct flood impact analysis in the

downstream area based on the monitoring of water level and rainfall in the whole drainage basin. That is, in the flood disaster prevention application, it is necessary to achieve both immediate processing and mass data processing.

Nevertheless, conventional IoT systems have been constructed as server-centric systems in which an IoT server on a cloud collects data from sensors or devices and controls them remotely [6]. Therefore, to apply an IoT system to an application for flood disaster prevention, it requires solving problems of server-centric IoT systems, such as increase of network load and delay of feedback control [7].

In this paper, we propose a distributed cooperative IoT system to resolve above problems for applications of flood disaster prevention. The proposed system implements an event-driven data collection mechanism to collect data efficiently on a Never Die Network to achieve disaster tolerance of the access network at the time of a disaster. Furthermore, a rule-based autonomous control mechanism is introduced as the edge computing in the IoT gateway. This mechanism issues a local alarm for flood even when the IoT gateway cannot be connected with the IoT cloud

* Corresponding author:

kitagami@tau.cugat.net (Shiji Kitagami)

Published online at <http://journal.sapub.org/ijit>

Copyright © 2016 Scientific & Academic Publishing. All Rights Reserved

server.

In this paper, we developed a prototype of the distributed cooperative IoT system. Then, in order to evaluate effectiveness of our proposed system, we conducted experiments using real data of the river in Japan. In addition, we conducted a field trial in Quang Nam Province, Vietnam. As the result of our evaluation, we have confirmed that up to 1/20, the proposed method can reduce the network load for flood monitoring, and can issue the flood warning at the appropriate time.

The remainder of this paper is organized as follows. Chapter 2 provides the background and the aim of our study. In Chapter 3, we propose a distributed cooperative IoT system for flood disaster prevention. Chapter 4 discusses the experimental results of our proposed system. Chapter 5 shows the results of a field trial in in Quang Nam Province, Vietnam. Finally, Chapter 6 concludes this paper and presents future work.

2. Research Background and Objectives

2.1. Application for Flood Disaster Prevention

Climatic change caused by global warming has been aggravated damage caused by floods by heavy rains all over the world [10]. To minimize flood disasters, it is indispensable to introduce a flood disaster prevention system that activates a flood warning to residents. Moreover, to issue a flood warning at proper timing, it is necessary to monitor not only water flow from upstream but also rainfall and water levels in the whole drainage basin (Figure 1).

However, recently frequent occurrence of locally heavy rain is causing the delay of a flood warning. For this reason, the study of flood forecasts based on geographic information and rainfall information in the drainage basin has been developed. For instance, Tanaka *et al.* modelled the runoff process and the flow process to the river of rainfall [10]. In addition, the Japan Meteorological Agency (JMA) has operated the alarm system of flood warnings using the runoff index computed from this model since 2008. In this way, for flood disaster prevention, the alarm system which issues a flood warning based on the analysis of water level data and rainfall in the whole drainage basin is required.

In addition, in countries where flood control measures such as levees and flood control channels are delayed, the flood disaster prevention system plays an important role. Nevertheless, the costs for system construction and operation inhibit the spread of flood disaster prevention systems [11].

2.2. Server-centric IoT System

By employing the IoT system that combines data analysis techniques, sensor technology and Internet technologies, it is expected the construction and operation of the flood disaster prevention system at low cost. Although IoT systems have been built for each application until now, the

IoT service platform to reduce system construction and operating costs has proposed [6]. The IoT service platform provides common functions for a variety of applications, such as data collection, remote control and device management (Figure 2). However, an IoT system on the IoT service platform is server-centric architecture. That is, an IoT cloud server collects data from sensors and devices, and controls them based on the analysis results of collected data. For this, in the server-centric IoT system, continual communication between IoT server and IoT gateway is needed for data collection and remote control. Here, an IoT gateway plays the role of protocol conversion and data exchange. For this reason, higher network load and delay of feedback control has become one of the most critical issues in the server-centric IoT system consisting of many sensors and devices [12]. Furthermore, when the connection between the IoT server and the IoT gateway is lost, the whole system won't function at all.

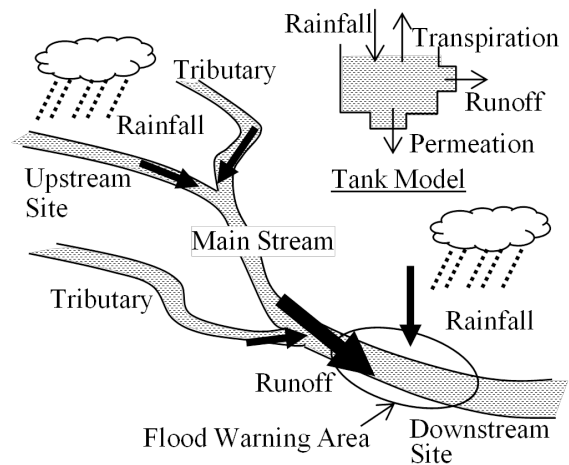


Figure 1. Flood Monitoring of Rivers and Tank Model

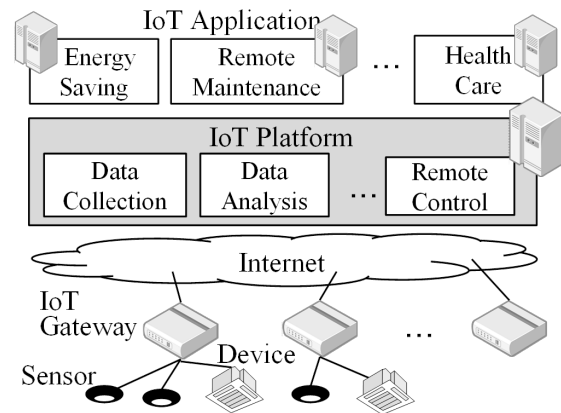


Figure 2. Server-centric IoT System

2.3. Research Objectives and Tasks

This study was undertaken to solve problems of a server-centric IoT system described above to extend the IoT system to prevent flood disaster. Here are three requirements to be fulfilled in this study.

- REQUIREMENT 1; The system for flood disaster prevention should not be interrupted even if communications infrastructure is partially damaged by flood or landslide.
- REQUIREMENT 2; The rainfall and water level data should be collected constantly because data are crucially important for flood forecasts, even if the network bandwidth is not sufficient.
- REQUIREMENT 3; Local flood warnings should be issued based on the monitored result of local rainfall or water level even if the network connection is lost.

3. Proposal of Distributed Cooperative IoT System

In this paper, we propose a distributed cooperative IoT system for implementing applications for flood disaster prevention satisfying the three requirements above.

3.1. System Configuration

Figure 3 shows the configuration of the distributed cooperative IoT system. In order to achieve disaster tolerance of the network for (REQUIREMENT 1), the distributed cooperative IoT system is constructed on a Never Die Network [8]. Also, we propose an event-driven data collection mechanism to realize efficient data collection in low-bandwidth networks for (REQUIREMENT 2). Furthermore, we introduce a rule-based autonomous control mechanism in an IoT gateway for (REQUIREMENT 3).

In Figure 3, an IoT coordinate server adjusts the roles of IoT gateway and IoT server for the application of flood disaster prevention. The IoT gateway that implements the proposed method will be referred to as an intelligent IoT gateway in this paper.

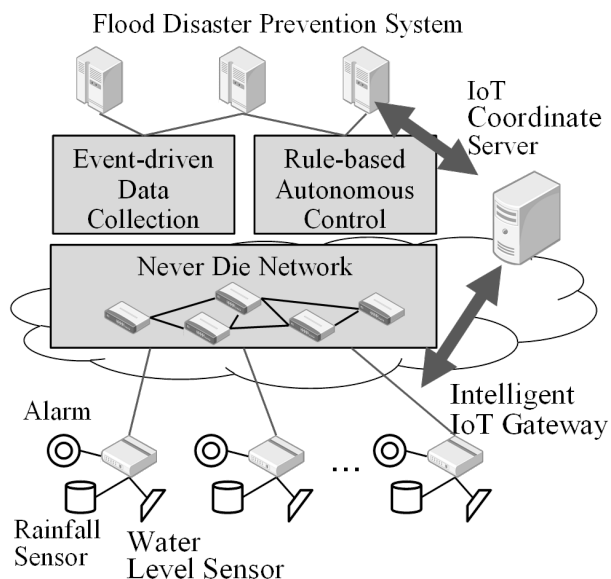


Figure 3. Proposed Distributed Cooperative IoT System

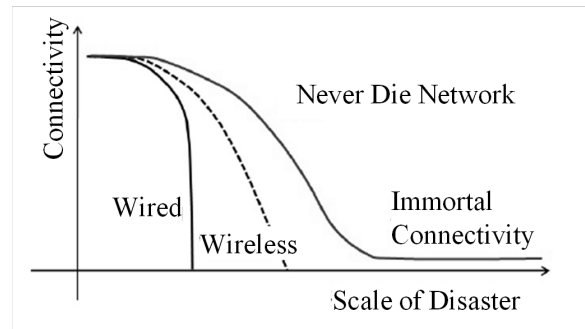


Figure 4. System Connectivity by Scale of Disaste [13]

3.2. Never Die Network

Loss of connectivity between an IoT server and an IoT gateway will cause of a system down and spoils the service continuity, as described in Chap. 2. Therefore, the distributed cooperative IoT system is based on the Never Die Network [8, 9, 13] to achieve disaster tolerance of the access network. The Never Die Network is constituted by cognitive radio nodes having a selectable plurality of links. Here, the network improves the disaster tolerance of the entire network by selecting the path or link based on throughput, latency and packet loss rate between the nodes. As shown in Figure 4, the Never Die Network keeps the data connection even if the transmission quality is lower than usual.

3.3. Event-driven Data Collection Mechanism

Conventional data collection methods in a server-centric IoT system include the following three methods.

- The serial data collection method; which collects data to each time measuring
- The cached data collection method; which collects data for each of the intervals
- The summarized data collection method; which collects only summary (average, etc.) of the measurement data for each of the intervals

The serial data collection method requires larger overhead of communication headers (IP protocol header, etc.) than measurement data (sensor data, etc.), thereby causes a heavy network load. Although the cached data collection method enables efficient data collection, large amounts of collecting data are unsuitable for a low bandwidth network environment. Moreover, this method is accompanied by a delay in feedback control based on the analysis result of collecting data on the IoT server. Both the serial data collection method and the cached data collection method must store all the data on the server, thereby requiring high data storage cost. The summarized data collection method can reduce both the network load and data storage cost in the server. However, the problem of feedback control delay would remain. In addition, this method cannot detect an abrupt change and singularity of measurement data. That is, the conventional data collection methods in a server-centric IoT system cannot guarantee the freshness and granularity of collecting data at the same time.

Event-driven data collection mechanism of a distributed cooperative IoT system automatically regulates the freshness and granularity of collecting data using variations in measurement data and related data. This mechanism enables the collection of data that changes rapidly while relieving the load of a network or a server storage cost. Furthermore, it avoids delay in feedback control based on the analysis result of collecting data.

Figure 5 shows the event-driven data collection mechanism of the distributed cooperative IoT system. The proposed mechanism summarizes measurement data at an intelligent IoT gateway, and sends the summarized data to the IoT server, and simultaneously holds original detailed data for a certain interval. It also monitors the variation in measurement data and associated data, and generates an event when the variation agrees with data variation pattern predefined. The gateway transmits the detailed data accumulated for a certain interval in addition to summarize data to the server upon the occurrence of this event. The conditions to generate an event for collecting detailed data can be selected from the following conditions.

- Threshold condition; when measurement data exceed a threshold
- Outlier condition; when an outlier to a moving average occurs
- Variation condition; when a change in measurement data is greater

Since only summarized data is collected by the event driven data collection mechanism, it is possible to reduce the amount of data collected in normal times, as similar to the summarized data collection method. Additionally, if the measurement data changes rapidly, it is possible to collect detailed data. Event conditions for data collection are transmitted from the IoT coordinate server in advance. This makes it possible to collect data under different conditions for each application.

By applying the event-driven data collection mechanism to applications for flood disaster prevention, it is possible to continue to collect the data required for flood forecasting, such as water level and rainfall, in a network of low bandwidth.

3.4. Rule-based Autonomous Control Mechanism

An application for flood disaster prevention issues a prior flood alert based on the results of monitoring water level and rainfall in the entire river area. It is also necessary to give real-time alerts of flash floods by local heavy rainfalls based on the results of monitoring water level and rainfall data in the area. However, in a server-centric IoT system, the feedback control to issue a flood alert for an evacuation of residents is delayed because of the increase in network load and server load. Moreover, when the connection between the IoT gateway and the IoT server is cut off, the application system loses its ability to function. We solved this problem using the edge computing technology.

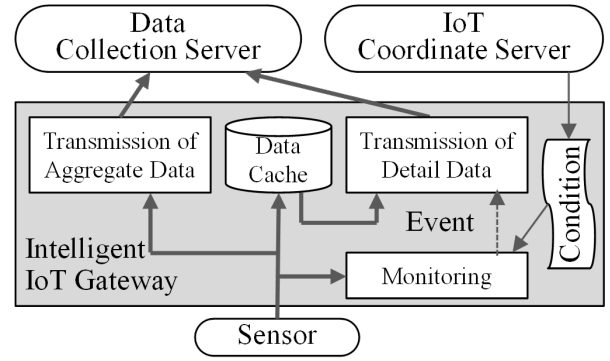


Figure 5. Event-driven Data Collection

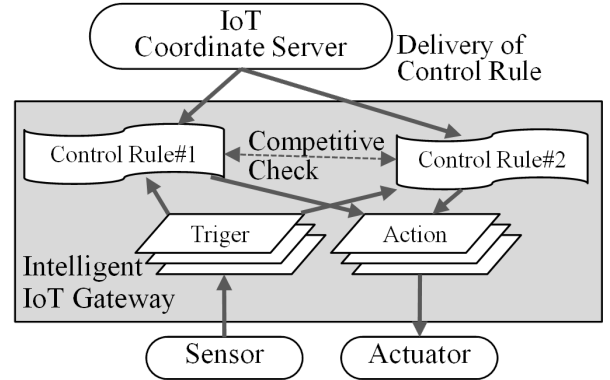


Figure 6. Rule-based Autonomous Control

<code>{"rule": "under_100_lx",</code>	<code>// rule name</code>
<code> "trigger": [</code>	<code>// trigger definition</code>
<code> {"url": "/sensor-01",</code>	<code>// sensor device</code>
<code> "name": "illumi-01",</code>	<code>// sensor name</code>
<code> "condition": "<100" },</code>	<code>// condition</code>
<code> "action": [</code>	<code>// action definition</code>
<code> {"url": "/actuator-01",</code>	<code>// actuator device</code>
<code> "name": "alarm-01",</code>	<code>// actuator name</code>
<code> "parameter": [</code>	<code>// control command</code>
<code> {"alert": "HIGH"}],</code>	<code>// priority</code>
<code> "priority": "100",</code>	<code>// valid duration</code>
<code> "duration": "60m" }</code>	

Figure 7. Example of Autonomous Control Rule

In rule-based autonomous control mechanism of a distributed cooperative IoT system, intelligent IoT gateway controls the connected device based on the measured data on the gateway. Figure 6 presents a rule-based autonomous control mechanism. In the mechanism, the control rules for autonomous control are sent in advance from an IoT coordinate server to an intelligent IoT gateway.

Figure 7 depicts examples of control rules for autonomous control by an intelligent IoT gateway. A control rule comprises actions and triggers. Actions describe a controlled object and its control method. On the other hand, triggers describe conditions for executing actions. An intelligent IoT gateway can accommodate the autonomous control rules of several applications.

By applying the rule-based autonomous control to applications for flood disaster prevention, an intelligent IoT gateway can issue a flood alert independently, even when an access network to IoT server is disconnected.

event-driven data collection mechanism and the rule-based autonomous control mechanism to the flood disaster prevention system.

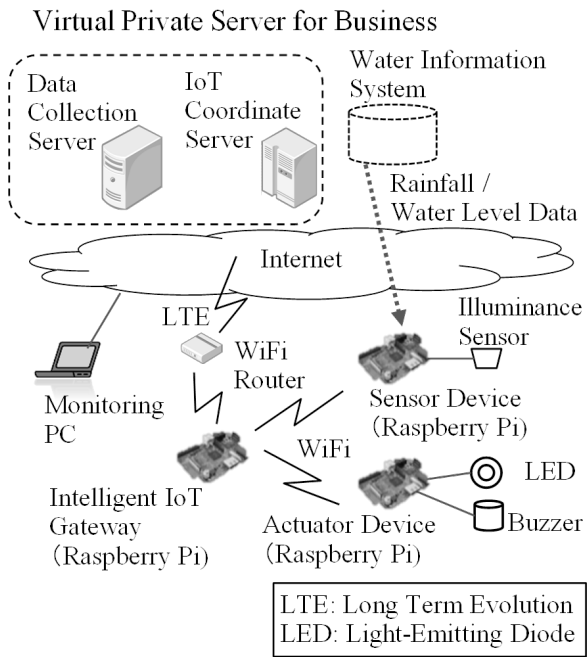


Figure 8. Prototype System for Evaluation

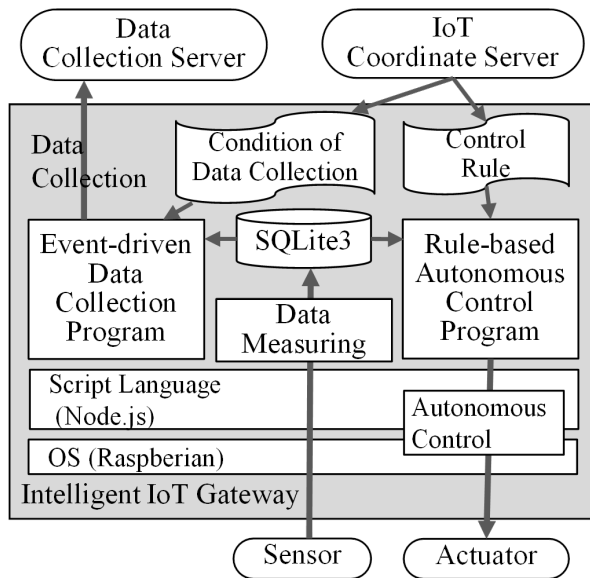
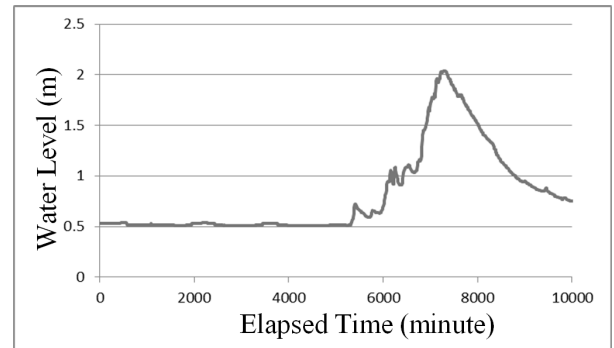


Figure 9. Intelligent IoT Gateway

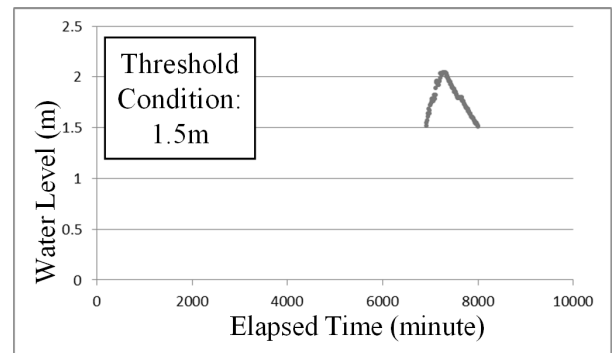
4. Evaluation and Discussion

4.1. Prototype System

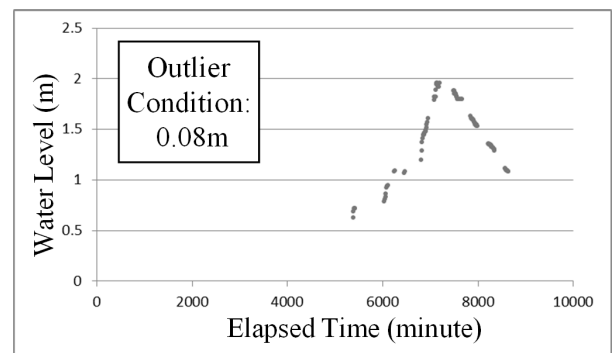
In this paper, we developed a prototype of the distributed cooperative IoT system operating on the usual Internet that does not include the Never Die Network. Then, using the prototype system, we evaluated the applicability of the



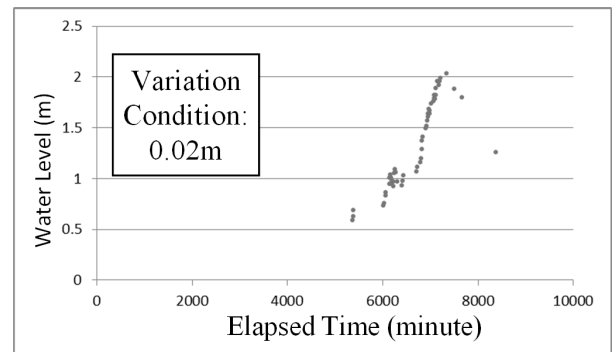
(a) Measured Original Data



(b) Threshold Condition



(c) Outlier Condition



(d) Variation Condition

Figure 10. Evaluation Results of Event-driven data collection

Figures 8 and 9 respectively show the entire configuration of the prototype system and the configuration of the intelligent IoT gateway. The data collection server and the IoT coordinate server were placed on the VPS (Virtual Private Server) of an actual commercial service. On the other hand, the intelligent IoT gateway, the sensor device and the actuating device were implemented on a single-board ARM microcomputer, Raspberry Pi. We adopted Node.js as a scripting language to implement the proposed method.

4.2. Event-driven Data Collection

A simulation was conducted for evaluation of event-driven data collection using water level data measured in the Asakawa River in Hachioji-shi, Tokyo (June 2–8, 2014, at a 10-minute interval) [14]. Figure 10 shows the results of the evaluation. The results of the evaluation show that the number of data collected by the proposed method is 110 in case of 1) the threshold condition. As well, it is 93 in case of 2) the outlier condition and it is 53 in case of 3) the variation condition. On the other hand, it is 1,008 in case of the conventional serial data collection method. Hence, the proposed method was able to reduce data to 1/10 -1/20 of the conventional method. In addition, the number of data is 181 in case of the combination of the above-mentioned three conditions. It becomes 1/6 of original data.

Event conditions in an event-driven data collection varies with applications. The threshold condition is a common condition, but it requires many data if values over a threshold appear frequently. The outlier condition is suitable for the application which data is no change in the overall trend but includes singular points. Applying the variation condition, it is possible to collect data relating to the signs of the trend changes, such as the rise and fall of the data. In applications for flood disaster prevention, the river water level rarely remains high, and the singularity is unlikely where the water level jumps up or down suddenly. Accordingly, the combination of threshold condition and variation condition is regarded as suitable for flood disaster prevention applications. That is, as shown in Figure 10 (b), when the water level exceeds a threshold, the system can collect detailed data by a threshold condition. Further, as shown in Figure 10 (d), when the water level started to rise, the system can collect detailed data by a variation condition.

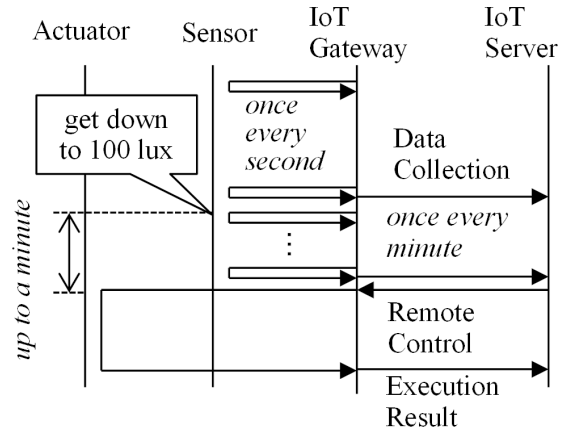
Although an event for the data collection is assigned to the change of the water level in this evaluation, it is possible to be assigned to the other data such as change in the rainfall. In that case, it is possible to collect detailed data of the water level only if the rainfall has exceeded the threshold value.

4.3. Rule-based Autonomous Control

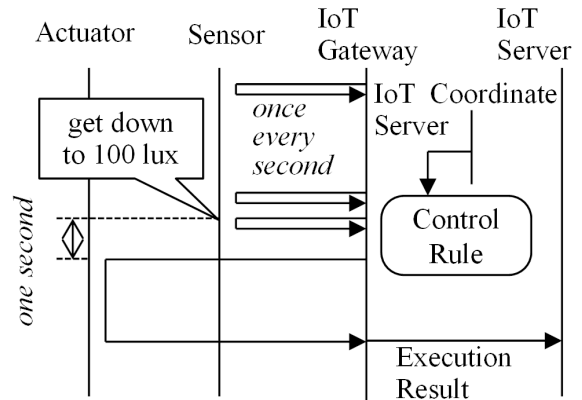
Figure 11 depicts the sequence flow of rule-based autonomous control mechanism implemented in an intelligent IoT gateway. In the prototype system, we substituted a rainfall and water level sensor with an illuminance sensor, and used an LED lamp and a buzzer as a substitute for a flood alert. In the cached data collection

method or summarized data collection method, which collects data for 1 minute interval, control delay is up to 60 sec. In contrast, employing the proposed method, it is possible to control the device in about 1 sec. at the same level as data measurement interval in the intelligent IoT gateway.

In applications for flood disaster prevention, adopting the rule-based autonomous control mechanism, even when the connection to the server is lost, it is possible to issue an alarm flood warning autonomously in response to changes in water level in the local area.



(a) Conventional Method



(b) Proposed Method

Figure 11. Sequence Flow of Rule-based Autonomous Control

4.4. Discussion of Applicability of Distribute Cooperative IoT System to Flood Disaster Prevention

The Distributed Cooperative IoT system is able to improve the availability of the flood disaster prevention system in three stages as below.

(Phase 1) If parts of the network equipment are damaged, and intelligent IoT gateway cannot connect to the IoT server, the Never Die Network re-establishes a connection with the IoT server by changing the communication path.

(Phase 2) If the network bandwidth cannot be sufficient, data required by the necessary minimum are collected by the event-driven data collection mechanism.

(Phase 3) If network connectivity is interrupted, the intelligent IoT gateway issues a flood alert autonomously by the rule-based autonomous control mechanism in response to changes in water level in the local area.

As described in Chapter 1, in flood forecasting, it requires rainfall data on the upstream, and it does not require the rainfall data downstream. That is, considering the shape of the river basin, the rainfall data can be segmented into each separated region, and can be collected effectively in applications for flood disaster prevention.

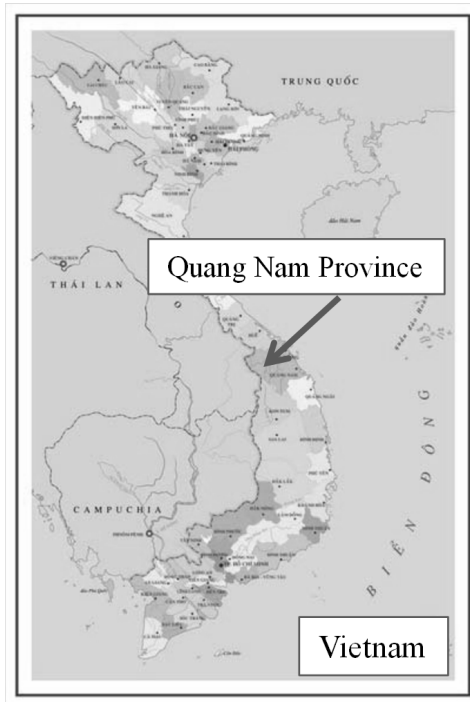


Figure 12. Location of Field Trial

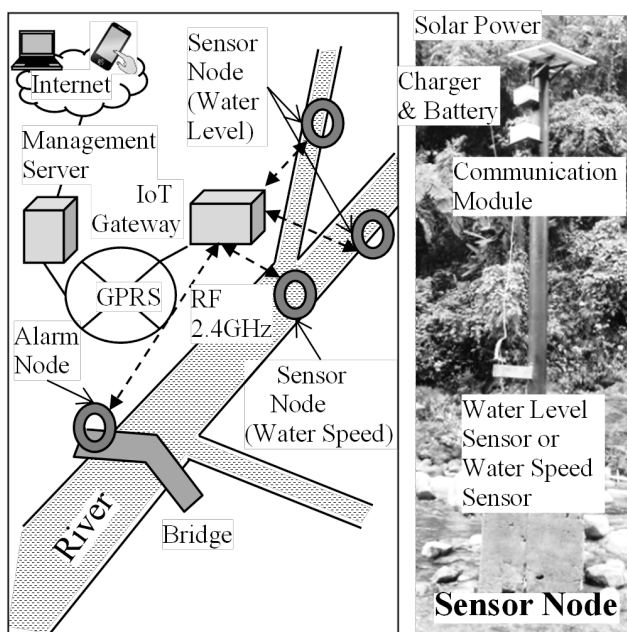


Figure 13. Field Trial System In Quang Nam Province, Vietnam

5. Field Trial in Quang Nam Province, Vietnam

In order to evaluate the effectiveness of the proposed method, we conducted a field trial of flash flood management in Quang Nam Province, Vietnam. The location of our field trial is shown in Figure 12. Quang Nam is located in the middle of Central Vietnam and is surrounded by Thua Thien-Hue, Quang Ngai, and Kon Tum provinces.

In recent decade, county of Vietnam is rapidly developing in even rural area [15, 16]. Thus, because of the loss of forest, there is a large rain storm in the mountain area; normally a flash flood will occur very often. Depending on the different heavy rains and the hilly terrains, the speed and the devastation of flash floods are also different. When the flash flood comes, there is almost no time for the people to evacuate. Unfortunately, at present there is no flash flood warning system in those places. As a result, local residents around these areas suffer from the heavy loss of houses, crops, cattle, money and even their lives. The outcome of our field trial of flash flood management is to provide the Local Government and the population with the needed tools to monitor and early warn the disaster.

The field trial focused on IoT solutions for flash flood management and early warning to the people around the areas. The installation and evaluation of the field trial system have been carried out from 31st Mar 2015 to 14th Apr 2015.

Figure 13 shows the configuration of the field trial system and a sensor node. The sensor measures the flow speed and water level at the upstream and stores the data with some interval. The data are collected on the placement of the sensor locations and send to a server. The IoT gateway receives data from sensor nodes over RF 2.4GHz, and sends data to the server over HTTP/GPRS. Also, the IoT gateway receives a command from the server to change data retrieval frequency, as well as warning activating command. Solar power includes 50 Wp solar panel and 10Ah VLRA battery. Hence the sensor node can work for 30 days, and the IoT gateway can work for 15 days.

A Management server analyses the information and gives warning alarm to the local governments and people to evacuate to safe places. By placing the sensor at the upstream, hopefully, the affected areas will be given more than 30 minutes of preparation time. The data are sent to the cloud for statistical collection for other assessments; so that the Local Government can have the damage estimation as well as the width of the danger areas. However, in the field trial, we could not detect a failed sensor node, which is installed in the mountain area. For the reason, it is necessary to introduce a mechanism to detect failed sensor nodes from data collected from other sensor nodes in the future.

6. Conclusions

We proposed a distributed cooperative IoT system for flood disaster prevention. The proposed system consists of an event-driven data collection mechanism on the Never Die Network and a rule-based autonomous control mechanism. The former makes it possible to collect data efficiently in low-bandwidth networks. And the latter makes it possible to issue real-time flood alert in local area even when the IoT gateway cannot be connected with the IoT server. In conclusion, using the distributed cooperative IoT system, we can reduce the damage level which the flood causes.

As future work, we have to study of the optimization of the data collection condition in various flood scenarios, and the failure detection method of sensor node from data collected from other sensor nodes. In addition, we have a plan to study of the collaborative method among intelligent IoT gateways for advanced edge computing.

ACKNOWLEDGEMENTS

Part of this work was carried out under the Cooperative Research Project Program of the Research Institute of Electrical Communication, Tohoku University, and the APT (Asia Pacific Telecommunity) J2 Project title: "Study of Flash flood Monitoring and Early Warning System using M2M and Cloud Computing Technology" in 2014.

REFERENCES

- [1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communication Surveys & TUTORIALS*, VOL.17, NO.4, 2015.
- [2] G. Wu, S. Talwar, K. Johnsson, and N. Himayat, "M2M: from mobile to embedded internet", *IEEE Communications Magazine*, Vol. 49, No. 4, pp. 36–43, 2011.
- [3] G. Lopez, "ENERSip: M2M-based platform to enable energy efficiency within energy-positive neighbourhoods", *Proceedings of Computer Communications Workshops (Infocom Wkshps)*, pp.217–222, 2011.
- [4] S. Jung, J.Y. Ahn, D. Hwang, and S. Kim, "An optimization scheme for M2M-based patient monitoring in ubiquitous healthcare domain", *International Journal of Distributed Sensor Networks*, Vol. 2012, Article ID 708762, pp.1–9, 2012.
- [5] The Ministry of Land, Infrastructure and Transport Japan, "Future of River Management in the Ubiquitous Information Society (recommendation)", <https://www.mlit.go.jp/common/000022382.pdf>, 2010 [accessed 12 April 2016].
- [6] D. Boswarthick, O. Iloumi, and O. Hersent, "M2M Communications: A Systems Approach," Wiley, ISBN: 978-1119994756, 2012.
- [7] S. Kitagami, Y. Myanishi, Y. Urano, and N. Shiratori, "Proposal of a Distributed Cooperative M2M System for Flood Disaster Prevention", *ScalCom-102, The 14th IEEE International Conference on Scalable Computing and Communications*, 2014.
- [8] N. Shiratori, N. Uchida, Y. Shibata, and S. Izumi, "Never Die Network towards Disaster-resistant Information Communication Systems," *ASEAN Engineering Journal Part D*, Vol.1, No.2, pp. 1-22, 2013 [Invited paper].
- [9] Y. Shibata, N. Uchida, N. Shiratori, "Analysis and Proposal of Disaster Information Network from Experience of the Great East Japan Earthquake," *IEEE Communications Magazine*, Vol.52, Issue 3, pp. 44-50, 2014 [Invited paper].
- [10] N. Tanaka, T. Oota, and Y. Makihara, "Flood warning/advisory improvement based on JMA Runoff Index", *Japan Meteorological Agency, Weather service bulletin*, Vol. 75, No. 2, pp. 35-69, 2008 (in Japanese).
- [11] M. Watanabe, "A Draft Proposal of a Plan for Disaster Mitigation and Preparedness applicable in Developing Countries", *Disaster Prevention Research Institute Annuals*, Kyoto University, No. 46, pp. 1-13, 2002 (in Japanese).
- [12] H. Takahashi, and T. Kushida, "Sensor Cloud: Extended Cloud Computing Environment to Use Sensor Devices as IT Resources", *Journal of Information Processing Society of Japan*, Vol. 54, No. 2, pp. 672-687, 2013 (in Japanese).
- [13] N. Uchida, K. Takahata, Y. Shibata, N. Shiratori, "Never die network based on Cognitive Wireless Network and satellite system for large scale disaster," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, Vol. 3, Num.3, pp. 74-93, 2012.
- [14] Water Information System, the Ministry of Land, Infrastructure and Transport Japan, <http://www1.river.go.jp/> [accessed 12 April 2016].
- [15] H. Q. Nguyen, J. Degener, and M. Kappas, "Flash Flood Prediction by Coupling KINEROS2 and HEC-RAS Models for Tropical Regions of Northern Vietnam," *Hydrology* 2015, Vol.2, Num.4, pp. 242-265. 2015.
- [16] C. Lu, J. V. Meding, S. Kanjanabootra, and D. Pham, "A proposed flood risk assessment method for Central Vietnam," *Proceedings ANDROID Residential Doctoral School: 5th International Conference on Building Resilience*, pp. 336 1-11, 2015.