Fast Recovery Architecture by Selecting Surrogate Master Node (FSMN) in Industrial IoT Environment

Taekyun Roh
Dept. Electronic & Computer Engn.
Sungkyunkwan University
Suwon, Korea
tkroh0198@skku.edu

Sooyeon Lee
Research & Development Center,
Gaesoft
Seoul, Korea
sylee@gaesoft.co.kr

Tai-myoung Chung
Dept. College of software,
Sungkyunkwan University
Suwon, Korea
tmchung@skku.ed

***Abstract*—This paper proposes an architecture that selects a sink node when a CPS is paralyzed. The CPS is a system (master node) with various sensor networks and machine networks that exchanges device information in real-time in the industrial IoT environment. When the CPS cannot perform normally due to internal/external factors, the last log of the master node is analyzed to determine the cause of the paralysis. The CPS select six attributes from the log data and convert them into datasets usable in this architecture. The normal category of the sink node is determined through the K-means clustering algorithm according to the sink node's profiling data. The centroid of the normal category is then determined. The normal category of the sink node is updated in real time until the CPS is paralyzed. When CPS paralysis occurs, the log data of the most recent sink node is applied to the K-means clustering that was formed in advance. We then determine the most available sink node among the sink nodes through the distance using vector value and priority algorithm between the central point and the sink node. Therefore, in the case of a failure of the master node, the high-priority surrogate master node can perform the role of the master node so that the data and the system can be maintained even in unexpected situations.**

***Keywords*— Industrial IoT, K-means Clustering, CPS, Fast Recovery Architecture, Priority Algorithm**

I. Introduction

Industrial internet-of-things (IoT) is based on the interaction of various Information & Communication Technology (ICT) with wireless sensor networks and communication technologies. In this environment, an extremely large amount of data is generated and consumed to provide services to people. These data are managed, controlled, and analyzed under a centralized architecture called the Cyber Physical System (CPS). The centralized data management system is useful as it integrates data physically and virtually. However, a problem arises when the system is paralyzed due to an unexpected situation, and the entire service stops.

The services provided in the case of a CPS failure are temporarily suspended. In addition, if a service interruption occurs in proportion to the failure recovery time, it may cause life-related problems or economic loss depending on the characteristics of the industrial environment to which the IoT system is applied.

In this paper, we consider the cause of the system failure through the K-means clustering algorithm. We then compare the sink node that can temporarily substitute the role of the CPS with the priority algorithm. The ultimate goal is to actively seek and select a surrogate master node to minimize the service paralyzed time. Therefore, we propose a method to prevent the loss of basic data by selecting the surrogate master node that can counteract the temporary paralysis of the CPS due to the specific cause through the Profiling, K-mean clustering algorithm, and the priority algorithm.

II. Related Work

The industrial IoT environment is composed of a combination of a Wireless Sensor Network (WSN), a Cyber Physical System (CPS), and a Machine to Machine (M2M) environment. M2M refers to the environment connected to machines, such as communication among objects and between humans and objects. WSN is an embedded wireless network technology that integrates all sensors in a network with wireless sensors connected to each other in real time. The WSN is connected to a computing environment regardless of its location.

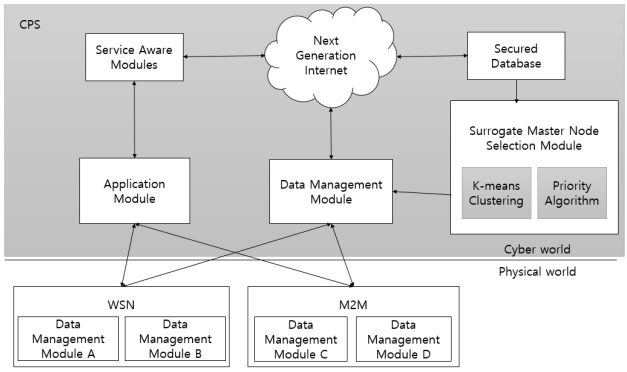


Fig. 1. Structure of CPS Using Surrogate Master Node Selection Module

IRJECE

This is an open access article under the CC BY-SA 4.0 license (https://creativecommons.org/licenses/by-sa/4.0/)

The WSN transfers data extracted from WSN to CPS. CPS is a real-time distributed control system that collects, analyzes, and controls data extracted from physical elements, connecting objects directly to the Internet without human intervention, and tightly combining and adjusting sensors and objects. Fig. 1 visualizes the communication between CPS, M2M, and WSN, which constitute an industrial IoT. [1].

III. Design Fast Recovery Atchitecture by Selecting a Surrogate Master Node(FSMN)

Before the system is paralyzed, the cause of the master node's failure must be determined in order to select the surrogate master node. Therefore, the priority of the surrogate master node should be determined through the data before the system suspension, and information about the surrogate master node should be periodically shared with the sink nodes. In the case of sudden system paralysis, we propose an architecture for fast system recovery through the surrogate master node's priority which was previously shared.

The master node is paralyzed in order to process the 6 attributes that can best determine the status of the master node among the log data of the master node and convert the 6 attributes into a profiling data composed of three attributes. After converting to the three profiling data, the resource with the lowest availability is judged as the main suspension cause. Profiling data refers to the process of selecting the required attribute values from the log data of the master node and converting them into a data format for application to this architecture.

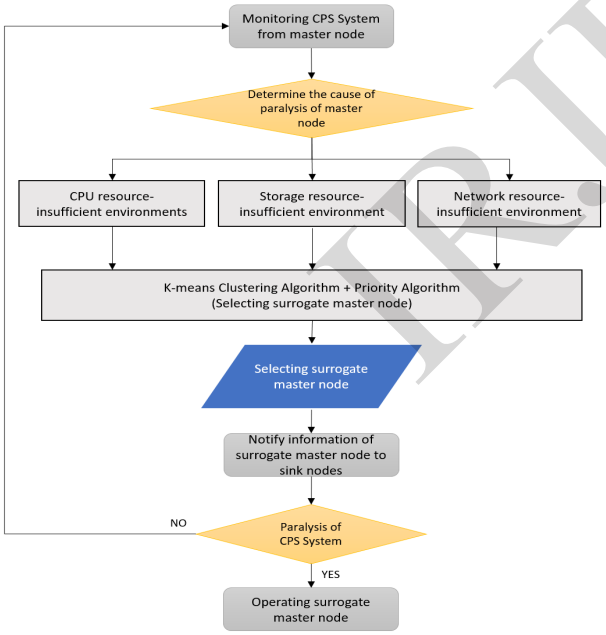


Fig. 2. Flow chart of Selecting Surrogate Master Node

Therefore, the cause of paralysis is identified through the profiling data, and the master node is selected by analyzing the available resources of the sink nodes. The master node selects the sink node by forming the profiling dataset through the log data collected from the sink node and applying it to the K-means clustering algorithm. When the surrogate master node is selected, the master node shares information about the surrogate master node with the sink nodes so that the surrogate master node selected at the time the master node is paralyzed can instantly be substituted for the master node. Through the above fast recovery architecture, the system paralyzing time can be minimized. Fig. 2 shows the process in which the surrogate master node is selected when the CPS is suspended.

A. Analysis of Log Data to Determine the Cause of Paralyzing the Master Node

The master node determines the possibility of paralyzing the master node through the log data of the master node. Based on the three related contents in the log data of the sink node, six attributes are calculated to determine the availability of central processing unit (CPU), storage, and network resources. Table 1 shows six attributes to be extracted from the log data.

TABLE I. 6-Attributes of Log Data

Content	Attributes
CPU Resource	Usage CPU Processing Rate
	Maximum CPU Processing Rate
Storage Resource	Usage Storage Space
	Maximum Storage Space
Network Resource	Usage Network Bandwidth Rate
	Maximum Network Bandwidth Rate

By converting the six attributes based on log data into three profiling data, the available ratio of each resource is calculated. Table 2 shows the availability ratio of the three resources, and the resource with the lowest ratio among the three resources is determined to be the cause of the system paralysis. Fig. 3 shows the pseudo code for generating the profiling data.

TABLE II. Profiling Data of Attributes

Content	Attributes
CPU Resource	Usage CPU Processing Rate / Maximum CPU Processing Rate (%)
Storage Resource	Usage Storage Space / Maximum Storage Space (%)
Network Resource	Usage Network Bandwidth / Maximum Network Bandwidth (%)

Algorithm 1: Generating the Profiling Data

Result: Prf_{dev}

1 **for** $Data_{Log}$ **do**
2 $CPU_{rate} = CPU_{usage} / CPU_{total}$
3 $Str_{rate} = Str_{usage} / Str_{total}$
4 $Net_{rate} = Net_{usage} / Net_{total}$
5 $Prf_{dev} = (CPU_{rate}, Storage_{rate}, Network_{rate})$(Profiling Data)
6 **end**
7 Prf_{dev}

Fig. 3. Pseudocode of Generating the Profiling Data

IRJECE

This is an open access article under the CC BY-SA 4.0 license (https://creativecommons.org/licenses/by-sa/4.0/)

B. Selecting a Surrogate Master Node Using K-means Clustering and Priority Algorithm

After determining the cause of the system suspension, the sink node selects a surrogate master node that is determined by the K-means clustering algorithm and can best serve the cause of the suspension. The log data of the sink nodes collected from the master node are profiled into three data sets shown in Section 3.1 and applied to the K-means clustering algorithm to generate three clusters. Each cluster generated by the K-means clustering algorithm represents a cluster of CPU resources, storage resources, and network resources. Fig. 4 shows a visual representation of the CPU resource, storage resource, and network resource cluster formation results from any of the 3,000 normal datasets.

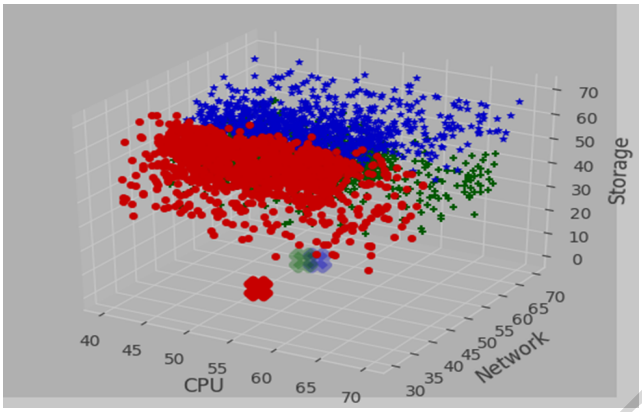


Fig. 4. Result of K-mean Clustering

The creation criteria of the dataset identify the normal category through the dataset under normal system operation. Therefore, it is randomly configured as a dataset of a sink node using 30% to 70% based on the availability of three normal data sets and the three resources applied in Fig. 4. The cluster with the normal range was formed through the data set constructed above.

In Fig. 4, '*' represents the CPU resource cluster, 'o' refers to the storage resource cluster, and '+' represents the network resource cluster. The centroid for each cluster is denoted by a bold X. After forming the cluster, the latest log data of the sink node is profiled and new profiling data is applied to the K-means clustering algorithm. The new profiling data is formed with one point in the K-means environment. The vector value between the sink node point formed to determine the resource suitability of the sink node and the Centroid corresponding to the cause of system suspension is calculated. The vector values of the distance to the centroid are represented by the component vectors for up to a newly formed sink node point and centroid. Fig. 5 shows vector values and component vectors up to a newly formed sink node point and centroid.

Vector a is the vector of the sink node point and centroid. ***(x_1, y_1, z_1)*** represents a direction vector for the x-axis, y-axis, and z-axis of ***Vector a.*** The x-axis represents the CPU resource, the y-axis represents the network resource, and the z-axis represents the storage resource. Therefore, the component vector and direction vector of all sink nodes generated from the latest log are based on the value obtained by converting the direction vector, which is the distance between the centroid of the cluster, into a scalar value. The master node selects the three sink nodes that are the farthest away from the positive direction vector of the cause of the master node paralysis. The farthest node is selected to ensure the sink node with the highest available resource.

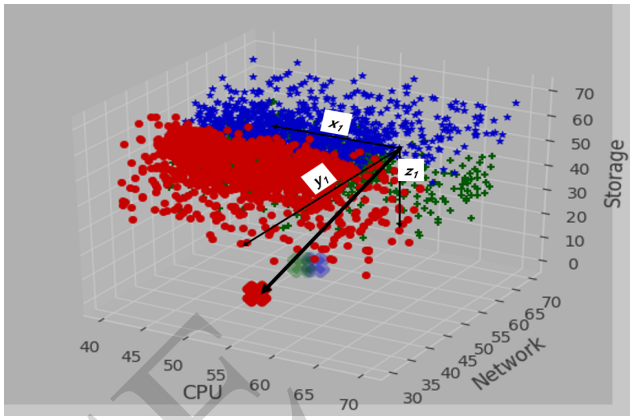


Fig. 5. Vector of Sink Node

The priorities of the previously selected sink nodes are determined using the priority algorithm for the component vector according to the paralyzed cause of the master node among the component vectors of each axis. The direction vector value of the cause of system suspension among the direction vector values of the previously selected sink nodes is selected. In addition, the direction vector values are converted into scalar values to prioritize three sink nodes. A sink node with a positive component vector means that the resource availability of the component vector axis is high, and a scalar value of the direction vector means that the entire resource is abundant. Therefore, the number of sink nodes is reduced by the component vector, and the surrogate master node is selected through the scalar value of the direction vector.

Algorithm 2: Selecting Surrogate Master Node

Result: $Prior1, Prior2, Prior3$

$Prf_{dev} = (CPU_{rate}, Storage_{rate}, Network_{rate})$

$\overrightarrow{dev}_{cause}(Com_{cpu}, Com_{str}, Com_{net}) = Cent_{cause}\text{-}Prf_{dev}$

$Scar(\overrightarrow{dev}) = \text{sqrt}(\overrightarrow{dev}^2_{cause})$

if $Com_{cause} \geq 0$ **then**

  for $3 \geq i$ **do**

    $Prioir_i = max(Scar(\overrightarrow{dev}))$ *(Choose 3 Max Scalar value)*

    i++

  end

end

Fig. 6. Pseudocode of Selecting Surrogate Master Node

This is an open access article under the CC BY-SA 4.0 license (https://creativecommons.org/licenses/by-sa/4.0/)

Fig. 6 shows a pseudocode representation of the process of determining priorities, and Table 3 describes the pseudocode terms.

TABLE III. DESCRIPTION OF TERMS

Term	Description
Prf_{dev}	Profiling dataset of 3 resources
Dev_{cause}	Dataset device according to system paralyzed cause
$Cent_{cause}$	Centroid of System paralyzed cause among generated clusters using K-means Clustering algorithm
$Scar(dev)$	Scalar value of device between centroid and profiling data
Com_{cause}	Compare value between centroid point and profiling point

C. Role of Surrogate Master Node depending on the Cause of Paralyzing the Master Node

CPS can become paralyzed when large amounts of computing and storage resources, and in high-bandwidth network environments are needed. We identify the cause of the paralyzed system and prioritize the surrogate master node through the K-means clustering algorithm and priority algorithm. The priority uses profiling data from the log data derived from the sink node. The profiling data uses the CPU resource usability rate, Storage resource usability rate, and Network bandwidth usability rate. According to the above three situations, the available resources of the sink node are identified, and the surrogate master node is ranked by the K-means clustering algorithm and the priority algorithm. CPS sends information to the sink nodes about the selected surrogate master node. Therefore, the surrogate master node is automatically selected through the priority of the sink node and data loss is prevented when the master node is paralyzed.

1) Computing resource-intensive Environment

In the industrial IoT environment, the available computing resources become insufficient when the data to be processed by the CPS system increases sharply and the consumption of computing resources is considerable. Thus, a reduced computing speed or paralyzing system can occur.

This situation causes data loss. In order to prevent loss of data, priority is given to the sink node according to the priority algorithm based on a node having a large amount of available CPU resource among the nodes. The surrogate master node substitutes the role of the master node until the computing capability of the master node is recovered.

2) Storage resource-intensive Environment

Data can be lost because of the limited storage space in the CPS system. When storage resource limitations arise, storage resources are supported by the surrogate master node. In this situation, the surrogate master node is selected by the sink node, which has the largest storage resources among the nodes. The node selected as the surrogate master node processes all the data until the available storage resources of the CPS are created. When the available storage space of the CPS system is recovered, the surrogate master node transmits the archived data back to the CPS to prevent data loss.

3) Network resource-intensive Environment

When the amount of traffic the CPS can receive is limited due to the limitation of the network bandwidth, a large amount of network traffic is transmitted and the packet is dropped. In order to prevent the packet from being dropped, the sink node is determined as the surrogate master node through the algorithm proposed in this architecture. The algorithm proposed in this paper measures the available network resources of the sink node to determine the most suitable surrogate master node. The surrogate master node receives and stores network traffic to be received by the CPS system. If the available network bandwidth of the CPS system is enabled for the current network conditions, the surrogate master node sends the network traffic back to the CPS system

VI. IMPLEMENTATION OF FAST RECOVERY ARCHITECTURE

1) Architecture for Determining the Cause of Master Node Paralysis

The master node can monitor the status of the master node through the above process. The logs generated by the master node include the usage/maximum CPU processing rate, the storage space rate, and the network bandwidth rate information of the master node. We convert the six attributes into profiling data for this architecture. The profiling data having the highest value is then selected as a basis for estimating the cause of the failure of the master node, and is shared with the information of the surrogate master node described in section 4.2

2) Atchitecture for Selecting a Surrogate Master Node

The suggested architecture for selecting a suitable surrogate master node is as follows. Profiling the logs that has been previously shared between the master node and the sink node is applied to K-Means Clustering, and then the log information transmitted from the sink node is profiled and coordinated. The log data sets used in this architecture are CPU utilization, network bandwidth, and available storage capacity. Since three datasets are used with six attributes in the log data, the log can be represented as a three-dimensional (3D) plane. In the 3D coordinate plane which expresses the result of applying K-Means Clustering, the X axis indicates the CPU, the Y axis indicates the network, and the Z axis indicates the storage. The number of clusters is three according to three resources. We consider that the surrogate master node has the highest availability through our proposed algorithm in this architecture.

3) Sharing the Information between Master Node and Sink Node

Candidate nodes of the master node and the surrogate master node share the following information in advance and respond to the failure of the master node. Table 4 shows the fields where the surrogate master node candidates selected from the master node are transmitted to the sink node.

The information that the Master Fault field means is the smallest value through the log of the master node and is used as a basis for estimating the failure cause of the master node. Surrogate Device id 1, 2, 3 contains the priority result of the surrogate master node selected from the cause of the paralyzed

IRJECE

This is an open access article under the CC BY-SA 4.0 license (https://creativecommons.org/licenses/by-sa/4.0/)

master node. For example, if a master node is suddenly paralyzed, the device of surrogate device id 1 becomes the surrogate master node. If surrogate device 1 cannot perform the role as surrogate master node because of the continuous role of the surrogate master node, then Surrogate Device 2 acts as next. In determining the paralysis of the master node, the surrogate master node is selected based on the information that is recently shared to cope with the causes of the master node paralysis.

TABLE IV. DATA FIELD OF BEING SELECTED SURROGATE NODE

Master Fault	Surrogate Device ID 1	Surrogate Device ID 2	Surrogate Device ID 3

VII. CONCLUSION AND FUTURE WORK

Currently, the industrial site on which the IoT system is built is rapidly increasing, and many existing industrial environments will need to change in order to adopt an IoT environment. Therefore, if the system paralysis occurs in the industrial IoT environment, it may cause various losses in proportion to the system paralysis time.

In the industrial IoT environment, the amount of data lost due to communication failure between networks caused by CPS failure increases in proportion to time. In addition, vast amounts of data are generated and controlled in real time. It creates an industrial value. Therefore, an architecture is needed that minimizes the value loss due to service delay through quick recovery even in the case of system failure.

In this paper, we propose a fast recovery architecture that reflects the environment of the industrial IoT system and a method to minimize the loss of data and service due to paralysis by selecting the surrogate master nodes to temporarily substitute CPS among all sink nodes.

In the future, we intend to implement a more reliable and analytical architecture through an analysis of the performance of this architecture by applying different clustering algorithms and applying other machine learning algorithms. We also intend to conduct research to improve accuracy through real dataset.

ACKNOWLEDGMENT

This work was supported by the Institute for Information & Communications Technology Promotion (IITP) with a grant funded by the Korean government (MSIP) No B0184-15-1003 and No R-20160222-002755, and the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (NRF-2010-0020210).

This research was supported by the MIST (Ministry of Science and ICT), Korea, under the National Program for Excellence in SW supervised by the IITP (Institute for Information & communications Technology Promotion)"(2015-0-00914).

REFERENCES

[1] Ahmed, Syed Hassan, Gwanghyeon Kim, and Dongkyun Kim, "Cyber Physical system: Architecture, applications And Research Challenges", In IEEE, 2017

[2] Soo Yeon Lee, Jun Kwon Jung,and Tai-myoung Chung, "A Study on the Network Failure Recovery for IoT Environment", The Korean Institute of Communications and Information Sciences Conference, pp194-195, 2017.

[3] Wu, Fang-Jing, Yu-Fen Kao, and Yu-Chee Tseng, "From wireless sensor networks towards cyber physical systems.", Pervasive and Mobile Computing 7.4, pp397-413, 2011

[4] J. A. Hartigan and M. A. Wong "Journal of the Royal Statistical Society. Series C (Applied Statistics)", Vol. 28, No 1, pp.100-108, 1979.

[5] "An Efficient k-means Clustering Algorithm: Analysis and Implementation", IEEE Transaction on Pattern Analysis and Machine Intelligence, vol 24, no 7, July, 2002

[6] "The Industrial Internet of Things(IIoT):the business guide to Industrial IoT", [online] Available at: https://www.i-scoop.eu/internet-of-things-guide/industrial-internet-things-iiot-saving-costs-innovation/, I-SCOOP, 2017

[7] Abolfazl Akbari, Arach Dana, Ahmed Khademzadeh, and Neda Beikmahdavi, "Fault Detection and Recovery in Wireless Sensor Network Using Clustering:, International Journal of Wireless & Mobile Networks, Vol3, No 1, Feburuary, 2011

[8] Zhexue Huang, "Extensions to the k-Means Algorithm for Clustering Large Data Set with Categorical Values", Data Mining and Knowledge Discovery 2,283-304, 1998

[9] Hong-chi Shih, Jiun-Huei Ho, Bin-Yih Liao, and Jeng-Shyang Pan, "Fault Node Recovery Algorithm for a Wireless Sensor Network", IEEE Sensor Journal, Vol 13, No 7, July, 2013

[10] Berihun Fekade, Taras Maksymyuk, Maryan Kyryk, and Minho Jo, "Probabilistic Recovery of Incomplete Sensed Data in IoT", IEEE Internet of Things Journal, Vol 13, No 7, July, 2013

[11] Hassen Redwan, Gebere Akele, and Ki-Hyung Kim, "Cluster-Based Failure Detection and Recovery Scheme in Wireless Sensor Network", The International Conference on Ubiquitous and Future Networks, 2014

[12] Jian Wan, Jianrong Wu, and Xianghua Xu, "A novel fault Detection and Recovery mechanism for Zigbee Sensor Networks", Second International Conference on Future Generation Communication and Networking, 2008

[13] Ameer A.Abbasi, Mohamed F.Younism and Uthman A.Baroudi, "Recovering From a Node Failure in Wireless Sensor-Actor Networks With Minimal Topology Changes", IEEE Transactions on Vihicular Technology, Vol 62, No 1, January, 2013

IRJECE

This is an open access article under the CC BY-SA 4.0 license (https://creativecommons.org/licenses/by-sa/4.0/)