Monitoring of IoT Data for Reducing Network Traffic

Jeongjin Lee, Gunjae Yoon, Hoon Choi
Department of Computer Science & Engineering
Chungnam National University
Daejeon, Republic of Korea
{likejj, yoongj, hc}@cnu.ac.kr

***Abstract*—** As IoT systems spread, a huge amount of data is generated from devices in every second. Transmission of all the IoT data to the remote OM(Operation and Management) server through the Internet may cause network traffic problems. In the paper, we propose an architecture of a monitoring node that uses the data pattern and monitors sensor data from IoT devices, so as to reduce network traffic and OM server's workload in IoT system environment. The monitoring node reports to the OM server only the data beyond the normal range of the pattern rather than sending all data to the OM server.

Keywords—Data Monitoring, Fog Computing, Network Traffic, IoT System

I. INTRODUCTION

Currently, Internet of Things(IoT)[1] is applied to intelligent home, smart building, factory automation, intelligent transportation system and autonomous car management system. The IoT system consists of a number of sensors, actuators, and computing nodes. Generally, these leaf-node devices do not have enough computing resources to supply intelligent service. Therefore, the processing, storing and classifying of the data are performed by one or more servers. Though a server or cluster of servers is able to process all the data generated from leaf-node devices, network problems such as traffic explosion, delayed transmission occur if the number of IoT systems increases. To alleviate this problem, CISCO introduced the concept of fog computing[2][3]. The fog computing is a technique that does not store a huge amount of data in a large data server, instead, the data is stored near the data source. The fog computing has the advantage of mitigating network traffic that is concentrated on the server.

The traditional fog computing systems simply performed filtered data transmissions to the OM server by using filtering criterion provided by the human administrator. The Smart Gateway[4] reduces the communication overhead of the core network and reduces the burden on the cloud. The human administrator of the Smart Gateway needs to change the filtering criterion when it is needed. On the other hand, the system proposed in this paper automatically sets up and updates the filtering criterion without administrator's intervention. The ADAptive Monitoring(AdaM)[5] is a framework for reducing the volume of generated data, as well as, network traffic between IoT devices and data management endpoints. AdaM uses both adaptive sampling and adaptive filtering algorithms and dynamically adapts the monitoring intensity and the amount of data disseminated through the network based on the current metric evolution. Unlike the monitoring node proposed in this paper, the AdaM runs on an IoT device itself. Therefore, it may be applied to the IoT nodes with some computing hardware.

In this paper, a better method of adaptive generation of data patterns and data monitoring for fog computing in order for reducing network traffic and workload of servers in IoT system environment is proposed.

II. DATA MONITORING SYSTEM

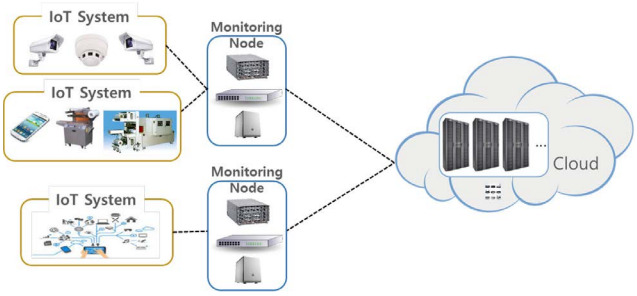

Fig. 1 Data Monitoring System

Fig.1 shows system architecture considered in this paper. The monitoring nodes connect several IoT systems with the OM server. A monitoring node is a proxy of the OM server and manages IoT system. To reduce network traffic, all the data generated by the IoT devices is transmitted to the monitoring node instead of the OM server. When a monitoring node receives a data, the monitoring node classifies this data whether to deliver to the OM server or not. This classifying process is performed according to the data pattern generated by the OM server. Due to this classifying process, the OM server receives data defined as being necessary only. The monitoring node has the following functions

A. Data storage

IoT devices of an IoT system may generate data frequently, possibly periodically. The data storage stores the data in a compressed form to reduce a storage capacity. When an unexpected or abnormal data is received, the data storage needs to record logs for system management and failure tracking.

The data storage requires efficient data management to avoid performance degradation. The monitoring node manages data on a block by block, for increasing the efficiency of data

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2017R1D1A1B03029262)

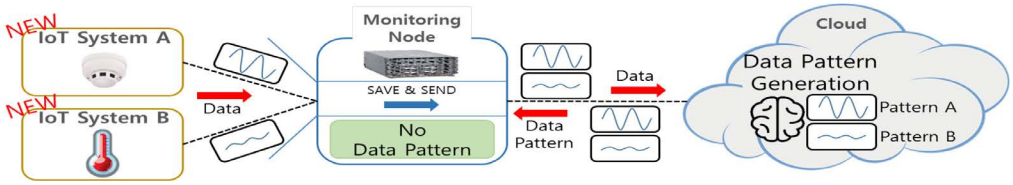

Fig. 3 Process for data pattern generation

search and transmission. A data set includes data values and a generation time with a device id and a sensor id for distinguishing which IoT device has generated the data. The data monitoring function also responds to the requests from the OM server. When the OM server requests specific data or data generated during the given time interval, the data monitoring function searches, and retrieves the data from the data storage and send them to the server.

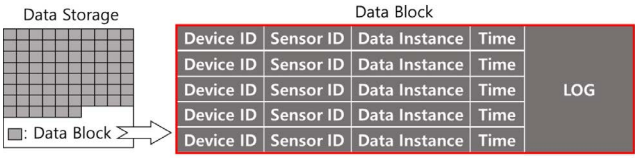

Fig. 2 Data block structure

B. Communication Management

This function transmits the data from/to an IoT device to/from an OM server. Each IoT system may use different communication protocols such as Modbus, MQTT[6] or CoAP[7]. IoT systems cannot be accessed from an OM server unless resolving this protocol difference. By converting legacy communication protocols between IoT devices and the monitoring node to a common, standard Internet protocol between the monitoring nodes and OM servers, the monitoring node can connect different IoT environments and therefore achieves the scalability. Data Distribution Service(DDS)[8] can be used for the common Internet-side protocol.

C. IoT Monitoring

This function monitors the connection statuses and operational statuses of connected IoT devices. The monitoring node collects hardware information about IoT device, location information such as an IP address, sensor information, and connected line status. For example, when a new IoT device is plugged into the IoT system, this function of the monitoring node automatically detects the device status and reports it to the OM server in real-time.

D. Data Monitoring

Date	Time	Mean Normal Value	Acceptable Deviation
2018-01-01	12:00:00	20.05	5%
2018-01-01	12:00:30	20.05	5%
2018-01-01	12:01:00	21.0	5%
2018-01-01	12:01:30	21.0	5%
:	:	:	:

Table. 1 An example of the data pattern for an IoT device recording normal values every 30 seconds

This function analyzes collected data. The monitoring node gets the data pattern from the OM server in order to judge the normality of the received data. The received data is saved by the data storage function. Then, the values of the received data are compared with the data pattern. If the data does not correspond to the normal data pattern, monitoring node records a log and reports the occurrence of abnormal data to the OM server. An example of the pattern may be as Table 1.

Generating a proper data pattern is an important subject. We assume that the OM server generates the data patterns and provides them to the monitoring nodes for each IoT device based on the big set of data collected by each IoT device. Therefore, technical issues on the data pattern are out of the scope of this paper.

III. Operational Process

Fig.3 shows the process for generating data patterns

A. Process for data pattern generation

(1) When a new IoT device is recognized by the monitoring node, the monitoring node reports to the OM server in real-time. (2) The IoT device transmits data to the connected monitoring nodes. The monitoring node stores the data received from the IoT device. (3) When the monitoring node receives a data in step (2), it also checks whether the data pattern for this device exists. If the data pattern does not exist, the monitoring node transmits the received data to the OM server for generating a data pattern. (4) The OM server learns the data from the monitoring node, and generates a pattern of this specific IoT device which is the collection of sets of the value and the generation time of the data from the device. The pattern tells us fluctuation of normal data values with respect to time of a day. (5) When a pattern is generated with sufficient data, for example, with data collected for several hours, the OM server sends the data pattern to the monitoring node.

B. Process for data monitoring using the data pattern

(1) The IoT device transmits data to the monitoring nodes. Monitoring node stores the data received from the IoT device. (2) The monitoring node also compares the IoT data with the data patterns at that time moment generated the OM server. (3) If the value lies within the acceptable deviation range, repeat B-(1) and B-(2). This range is set by the administrator. For example, 5% range means that +/- 5% of the difference between the received data value and pattern value is accepted as a normal case. (4) If the value is beyond the normal data pattern range, the monitoring node records a log. The monitoring node also reports to the OM server of this abnormal value with time information. (5) The OM server recognizing the abnormal situation can analyze the data and alert the administrator or may reflect the data to update corresponding data pattern. The server may request the monitoring node of stored data of specific time interval for operation and management purpose.

Table.2 shows operational processes related to the monitoring node functions defined in section 2.

Monitoring Node Function	Operational processes
IoT monitoring	A-(1)
Data store	A-(2), B-(1)
Data comparison	B-(2), B-(3)
Converting to a real-time a communication protocol	A-(1), A-(3), A-(5), B-(4)
Generate data pattern	A-(4)
Analyze data pattern	B-(2)
Data report	B-(4)
Data request	B-(5)

Table. 2 The scenarios related to the monitoring node functions

Advantages of the proposed method compared with other fog computing mechanisms include that the data filtering is performed by a per-device base, by having data pattern for each IoT device. Unlike other mechanisms, our method does not require a human effort to provide data filtering criterion (normal data patterns), the data patterns are made systematically and provided by the server. The data patterns can dynamically adapt appropriate situations, for example, different versions of data patterns can be made with respect to time progress of a day, seasonal or climate change.

IV. Performance Evaluation

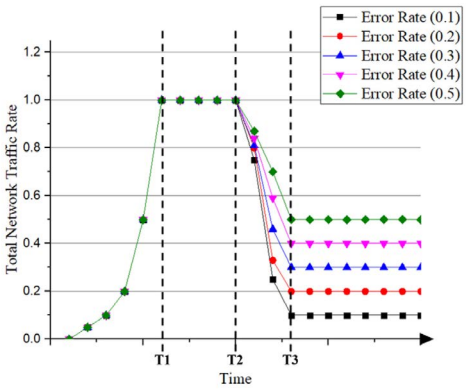

Fig. 4 Total network traffic ratio over time

We performed a simple evaluation in order to show the effectiveness of the proposed design of this paper. Fig.4 shows the ratio of network traffic decrease obtained by simulation. From time 0 to T1, a monitoring node discovers and establishes the connection with the new IoT devices. During this time span, the network traffic is gradually increased as the new device begins transmitting data. When it reaches the time T1, discovering and connecting processes are completed. From time T1 to T2, the monitoring node transmits all the data to the OM server on the Internet because the monitoring node does not have a data pattern yet for this device. Therefore, the network traffic reaches the maximum and a workload of the OM server is also high. During this period the OM server learns and generates the patterns from the receiving data. At time T3, pattern generation is accomplished and it is sent to the monitoring node. After T3, only the abnormal data is sent to the server.

The network traffic occurs in proportion to the assumed error rate, the rate of occurrence of abnormal data from some IoT device. In a typical IoT system without the monitoring node, all the data from IoT devices is directly sent to the OM server, the network traffic ratio will always be 1 after T1. If we assume the error rate to be 0.2, 80% of network traffic are not sent to the OM server. Therefore, the proposed method can decrease network traffic a lot than typical IoT system.

V. Conclusion

In this paper, we proposed an architecture of a monitoring node functional design level. The purpose of the monitoring node is to reduce network traffic and server's workload in IoT system environment. Monitoring of normality of data from each IoT device is based on its data pattern provided by the OM server. The monitoring node with the data patterns reports to the OM server only the data beyond the normal range rather than sending all data to the OM server.

Implementation of the monitoring node is currently ongoing. We plan to measure detailed performance of the system with respect to the network traffic reduction and throughput of the monitoring node.

Acknowledgment

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2017R1D1A1B03029262).

References

[1] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, Marimuthu Palaniswami. "Internet of Things(IoT): A vision, architectural elements, and future directions." Future generation computer systems, 29.7, pp.1645-1660, 2013.

[2] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, Sateesh Addepalli. "Fog computing and its role in the internet of things." In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM, pp.13-16, 2012.

[3] Ivan Stojmenovic. "Fog computing: A cloud to the ground support for smart things and machine-to-machine networks." In: Telecommunication Networks and Applications Conference (ATNAC), 2014 Australasian. IEEE, pp.117-122, 2014.

[4] Mohammad Aazam, Eui-Nam Huh, "Fog Computing and Smart Gateway Based Communication for Cloud of Things." Future Internet of Things and Cloud, pp.464-470, 2014.

[5] Demetris Trihinas, George Pallis, Marios D. Dikaiakos, "AdaM: an Adaptive Monitoring Framework for Sampling and Filtering on IoT Devices." 2015 IEEE International Conference on Big Data(Big Data), pp.717-726, 2015.

[6] Sung-Jin Kim, Kyoung-Woo Cho, Myung-Eui Lee, Chang-Heon Oh. "A Study on Adaptive QoS Control System based on MQTT for Reducing Network Traffic." INTERNATIONAL CONFERENCE ON FUTURE INFORMATION & COMMUNICATION ENGINEERING, 9.1, pp.223-226, 2017.

[7] Shelby, Zach, Klaus Hartke, and Carsten Bormann. "The constrained application protocol (CoAP)." 2014.

[8] Pardo-Castellote, Gerardo, Bert Farabaugh, and Rick Warren. "An introduction to DDS and data-centric communication." 2005 Real-Time Innovations, 2005.