Journal of Computational Science 22 (2017) 171–178

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Adaptive data rate control in low power wide area networks for long range IoT services

Dae-Young Kim[a,1], Seokhoon Kim[b,1], Houcine Hassan[c], Jong Hyuk Park[d,*]

[a] *Dept. of Computer Software Engineering, Changshin University, Changwon, Republic of Korea*
[b] *Dept. of Computer Software Engineering, Soonchunhyang University, Asan, Republic of Korea*
[c] *Dept. of Computer Engineering, Universitat Politecnica de Valencia, Valencia, Spain*
[d] *Dept. of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul, Republic of Korea*

ARTICLE INFO

Article history:
Received 31 August 2016
Received in revised form 5 April 2017
Accepted 21 April 2017
Available online 24 April 2017

Keywords:
LPWAN
IoT
Adaptive data rate control
Congestion identification
Data transmission

ABSTRACT

Internet of Things (IoT) technologies can provide various intelligent services by collecting information from objects. To collect information, Wireless Sensor Networks (WSNs) are exploited. The Low Power Wide Area Network (LPWAN), one type of WSN, has been designed for long-range IoT services. It consumes low power and uses a low data rate for data transmission. The LPWAN includes several communication standards, and Long Range Wide Area Network (LoRaWAN) is the representative standard of the LPWAN. LoRaWAN provides several data rates for transmission and enables adaptive data rate control in order to maintain network connectivity. In the LoRaWAN, the wireless condition is considered by the reception status of the acknowledgement (ACK) message, and adaptive data rate control is performed according to the wireless condition. Because the judgment of the wireless condition by the reception status of ACK messages does not reflect congestion, adaptive data rate control can lead to inefficiency in data transmission. For efficient data transmission in long-range IoT services, this paper proposes a congestion classifier using logistic regression and modified adaptive data rate control. The proposed scheme controls the data rate according to the congestion estimation. Through extensive analysis, we show the proposed scheme's efficiency in data transmission.

1. Introduction

Internet of Things (IoT) has become an important computing paradigm. It provides network connectivity among objects, small embedded systems that include sensing, computing, and networking capabilities. The objects collect sensing information and deliver it to a network cloud. A network server in the cloud receives and analyzes the objects' information. Based on the analyzed information, an application server can provide new intelligent services as IoT services. Examples of IoT services include smart home, smart transportation, smart agriculture, smart factory, etc. The IoT services, are called as Cyber Physical Systems (CPS), connect the collected information from the objects in the physical world to perform decision making in the computing world. Because IoT services depend on collecting information from objects, effective information gathering is important. Thus, Wireless Sensor Network (WSN) technologies are exploited to collect information from objects in IoT [1–7]. Fig. 1 shows the IoT network architecture.

Conventional WSNs have dealt with the access network for IoT services. The WSNs, which are composed of sensors and actuators as objects, have been developed according to the point of view in personal area networks. They collect information from objects using short-range communication technology [8–10]. Thus, the necessity of long-range communication technology in WSNs has appeared for various IoT services in wide areas, and Low Power Wide Area Network (LPWAN) technologies have emerged. LPWAN IoT devices consume low transmission power and use a low data rate. In addition, they have a communication distance of several kilometers. The LPWAN is considered a wide area IoT sensor network [11–17]. Among the LPWAN technologies, the Long Range Wide Area Network (LoRaWAN) is the representative standard, and it is widely exploited in various areas.

Although the LoRaWAN uses a low data rate, it provides adaptive data rate control in order to maintain network connectivity. The RF module of the LoRaWAN supports a data rate of several steps. Adaptive data rate control is related to the modulation coding

[*] Corresponding author.
E-mail addresses: kimdy@cs.ac.kr (D.-Y. Kim), seokhoon@sch.ac.kr (S. Kim), husein@disca.upv.es (H. Hassan), jhpark1@seoultech.ac.kr (J.H. Park).
[1] These authors contributed equally to this work.

http://dx.doi.org/10.1016/j.jocs.2017.04.014
1877-7503/© 2017 Elsevier B.V. All rights reserved.

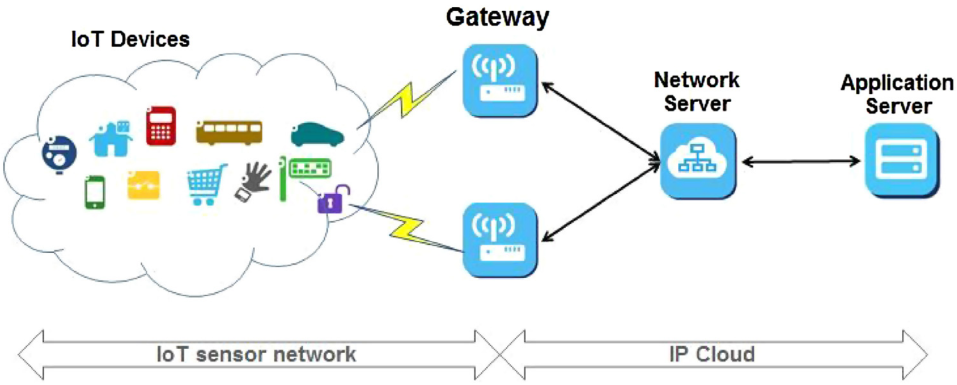


Fig. 1. IoT network architecture.

scheme of the RF module. That is, the modulation scheme or coding rate of the RF module changes according to the data rate step. In general, a lower data rate has a longer transmission distance. Thus, when there is a connectivity problem, the LoRaWAN chooses the lower data rate and tries to reconnect. This is an important behaviour in long-range IoT services, because data is transmitted across long distances with low transmission power, and frequent connectivity failures can occur during data delivery. The existing LoRaWAN scheme judges the status of the wireless connection through the reception status of the acknowledgement (ACK) message [18]. Because this does not reflect network congestion, it can lead to inefficiency in data transmission. If the cause of the reception failure of the ACK message is the poor condition of the wireless link, dropping the data rate can effectively extend network coverage and maintain the wireless link. However, if the cause is network congestion, dropping the data rate can reduce transmission efficiency and lead to long transmission delays.

In an LPWAN, a network constructs a star topology because of the long transmission range. Numerous IoT devices are deployed to provide services. Thus, congestion conditions can occur frequently. Because congestion is not a connectivity problem, lowering the data rate by changing the modulation scheme is not the appropriate choice. Therefore, in this paper, the proposed method tries to avoid inappropriate data rate control through learning by logistic regression. The proposed method exploits the data rate, received signal strength, and number of connections at a gateway as data attributes for learning. Using the estimation results of the reason for the reception failure of the ACK message, the proposed method performs adaptive data rate control. With the proposed method, long-range IoT services can provide transmission efficiency in IoT sensor networks.

As mentioned earlier, data rate control is an important feature to adjust communication coverage. The current LoRaWAN depends on the reception of ACK messages to control the data rate. This causes an unexpected result: that is, data rate control can provide communication efficiency in wireless conditions with link errors (not congestion). Therefore, the proposed method recognizes the congestion status of the network by learning and applies the learning result to data rate control. This improves the accuracy of the data rate control and the network efficiency for LoRaWAN.

The remainder of this paper is organized as follows. Section 2 describes the LoRaWAN and the existing adaptive data rate of the LoRaWAN as related work. In Section 3, learning on wireless status using logistic regression and the modified adaptive data rate control method are discussed. In Section 4, a performance analysis is carried out by comparing the proposed scheme to the existing LoRa scheme. Finally, Section 5 concludes the paper.

2. Related work

WSNs are necessary to implement IoT services. They connect objects and collect information from the objects. Existing WSNs focus on a personal operating space that extends up to 10 m in all directions [19,20]. That is, their service area is very small. Thus, for various IoT services, a long-range sensor network technology (i.e., the LPWAN) is required. Because the LPWAN enables the collection of data in wide areas, wide area applications (e.g., smart city, smart agriculture, smart environmental monitoring, etc.) can be served easily.

In the LPWAN, there are several *de facto* standards: LoRaWAN, SigFox, and Weightless [12–17]. They have very similar characteristics and specifications, and they deal with low-data rate, low-cost, and low-power communication while focusing on long-range communication. Among them, the LoRaWAN is the representative standard. The LoRaWAN is composed of end-devices as objects, a gateway, and a network server. As shown in Fig. 2, the end-devices include the slave function and the network server includes the master function. Thus, the network server collects the information of the end-devices and manages the end-devices. The gateway relays data from the end-devices to the network server.

The LoRaWAN employs an AES 128-bit key mechanism for security. LoRaWAN devices have a 128-bit application key, which is provided by an application provider. They derive a network session key and an application session key using the application key. When a device receives a join-accept message during the network join procedure, it receives a random code from the network server. Using the random code and the application key, the device derives the session keys. Then, MAC (i.e., Medium Access Control) data messages are encrypted by the network session key and application data messages are encrypted by the application session key. Each device has a unique session key and provides security by delivering the encrypted messages using the keys.

The LoRaWAN supports adaptive data rate control to maximize network capacity. When an unreliable wireless link exists, adaptive data rate control is used to maintain the wireless link. For adaptive data rate control, an end-device periodically validates that it can receive an ACK message from a gateway. The end-device has a counter (*ADR_ACK_CNT*), and it adds one each time it transmits uplink data. The value of the counter reaches the threshold without any reception of ACK messages, and then it sets the ACK delay time (*ADR_ACK_DELAY*) to adjust the data rate and waits for the ACK messages. If no reply is received within the time, the end-device tries to regain connectivity by switching to the next lower data rate. That is, the end-device controls the data rate through the reception status of the ACK messages [18]. Fig. 3 represents the adaptive data rate of the LoRaWAN.

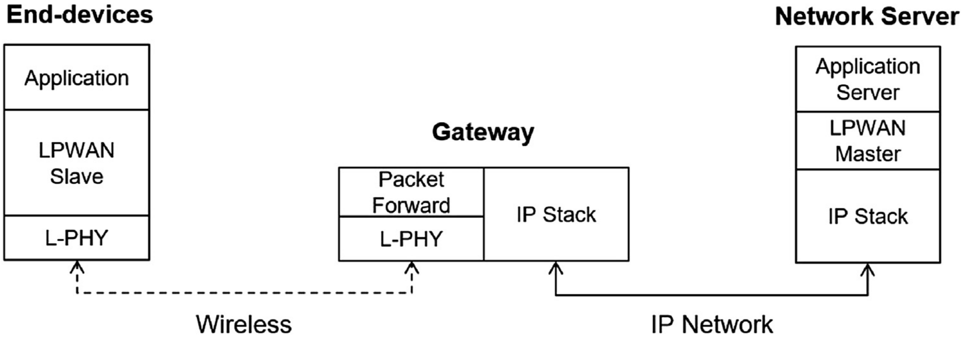


Fig. 2. LoRaWAN network architecture [18].

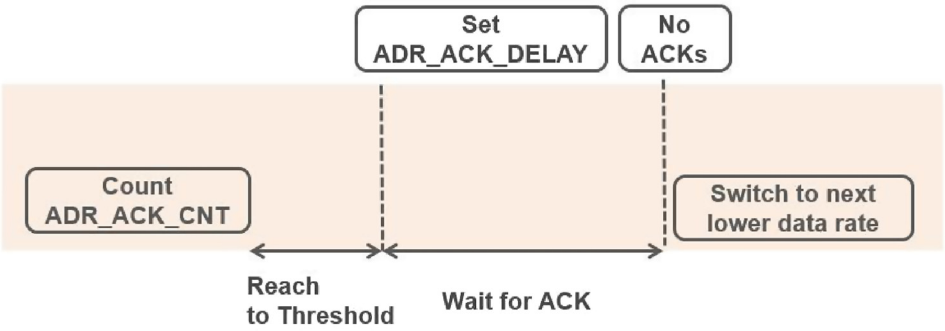


Fig. 3. Adaptive data rate control of LoRaWAN.

As mentioned earlier, the existing method of data rate control does not reflect network congestion. The data generated from objects is not stream data, so the end-device tries to change the RF modulation scheme to adjust the data rate. Thus, expanding connectivity by changing the data rate (i.e., RF modulation scheme) leads to inefficiency in data transmission. The proposed method tries to avoid this problem by estimating the congestion status of the wireless link before changing the data rate.

3. The proposed adaptive data rate control method

The proposed method is composed of the congestion classifier and the modified data rate controller. The congestion classifier uses the following statistical data of a network: throughput (x_1), received signal strength (x_2), and number of connections at a gateway (x_3). It learns through logistic regression. Using the results, it determines the congestion status (Y). Then, adaptive data rate control (DR) is performed according to the congestion status. Fig. 3 shows the system architecture for the proposed method.

3.1. Congestion classifier

The congestion classifier estimates network congestion through supervised learning. Supervised learning is widely used for status estimation in wireless networks. Status estimation is applied to make network algorithms for efficient data transmission [21–24]. The proposed method employs logistic regression to estimate the congestion status. The logistic regression classifies binomial states using a sigmoid function. The congestion status can be represented as a binomial distribution:

$$Y \in \{0, 1\}. \tag{1}$$

If Y has 1, the network experiences congestion. Otherwise, it does not.

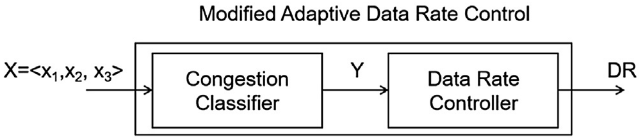


Fig. 4. System architecture of the proposed method.

Hypothesis ($h(X)$) for the status estimation is described as a sigmoid function, which is a logistic function that ranges between 0 and 1. The hypothesis can easily divide the binomial states using mathematics.

$$h(X) = g(\theta^T X) = g(z) = \frac{1}{1+e^{-z}}. \tag{2}$$

In the sigmoid function, the input parameter (z) is represented to a linear function of the data attributes (x) and weights (θ) as follows:

$$\begin{aligned} z &= \theta^T X = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n, x_0 = 1 \\ &= \theta_0 + \sum_{j=1}^{n} \theta_j x_j . \end{aligned} \tag{3}$$

As shown in Fig. 4, because the proposed method exploits three attributes (i.e., data rate, received signal strength, and number of activated connections at a gateway) as statistical data X, n in Eq. (2) is 3.

Given the logistic regression model with θ, the probability of network congestion is

$$P(Y = 1|X; \theta) = h(X). \tag{4}$$

The probability that network congestion does not occur is

$$P(Y = 0|X; \theta) = 1 - h(X). \tag{5}$$

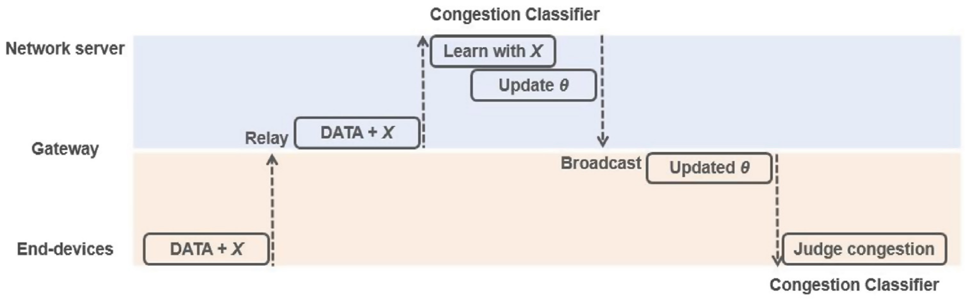


Fig. 5. Learning system.

Therefore, the probability of the congestion status becomes

$$P(Y|X;\theta) = (h(X))^{Y}(1-h(X))^{1-Y}, \tag{6}$$

where y represents the congestion status with 1 or 0.

If m training examples for learning about the congestion are generated independently, the probability of Eq. (6) can change to a likelihood function on θ.

$$\begin{aligned} L(\theta) &= P(Y|X;\theta) \\ &= \prod_{i=1}^{m} P(y^{(i)}|x^{(i)};\theta) \\ &= \prod_{i=1}^{m} (h(x^{(i)}))^{y^{(i)}}(1-h(x^{(i)}))^{1-y^{(i)}}. \end{aligned} \tag{6}$$

The likelihood function can also represent a log likelihood function as follows:

$$\begin{aligned} l(\theta) &= \log L(\theta) \\ &= \sum_{i=1}^{m} \left[y^{(i)} \log h(x^{(i)}) + (1-y^{(i)}) \log(1-h(x^{(i)})) \right]. \end{aligned} \tag{7}$$

Then, the congestion classifier judges the congestion by θ to maximize the log likelihood function. Finding the appropriate θ is the learning performed by the congestion classifier. To find θ to maximize the log likelihood function, the gradient ascent optimization algorithm is exploited. The gradient ascent algorithm for the congestion classifier is represented as follows:

$$\theta := \theta + \alpha \nabla_{\theta} l(\theta). \tag{8}$$

θ is updated according to the gradient of the log likelihood function. α is the unit size for the gradient. A derivative of the sigmoid function is as follows:

$$g'(z) = g(z)(1-g(z)). \tag{9}$$

Thus, the gradient of the log likelihood function at the ith training example is represented as follows:

$$\begin{aligned} \frac{\partial}{\partial\theta_j} l(\theta) &= \left(y^{(i)} \frac{1}{g(\theta^T x^{(i)})} + (1-y^{(i)}) \frac{1}{1-g(\theta^T x^{(i)})}(-1) \right) \frac{\partial}{\partial\theta_j} g(\theta^T x^{(i)}) \\ &= \left(y^{(i)} \frac{1}{g(\theta^T x^{(i)})} + (1-y^{(i)}) \frac{1}{1-g(\theta^T x^{(i)})}(-1) \right) g(\theta^T x^{(i)})(1-g(\theta^T x^{(i)})) \frac{\partial}{\partial\theta_j} \theta^T x^{(i)} \\ &= \left(y^{(i)}(1-g(\theta^T x^{(i)})) - (1-y^{(i)}) g(\theta^T x^{(i)}) \right) x_j^{(i)} \\ &= (y^{(i)} - h(x^{(i)})) x_j^{(i)}. \end{aligned} \tag{10}$$

Then, the updated weight θ becomes

$$\theta_j := \theta_j + \alpha(y^{(i)} - h(x^{(i)}))x_j^{(i)}. \tag{11}$$

By the stochastic gradient ascent algorithm, θ in the m training examples is represented as

$$\begin{aligned} &repeat\,\{ \\ &for\; i = 1\, to\, m\,\{ \\ &\theta_j := \theta_j + \alpha(y^{(i)} - h(x^{(i)}))x_j^{(i)},\ (\text{for every} j). \\ &\} \\ &\} \end{aligned} \tag{12}$$

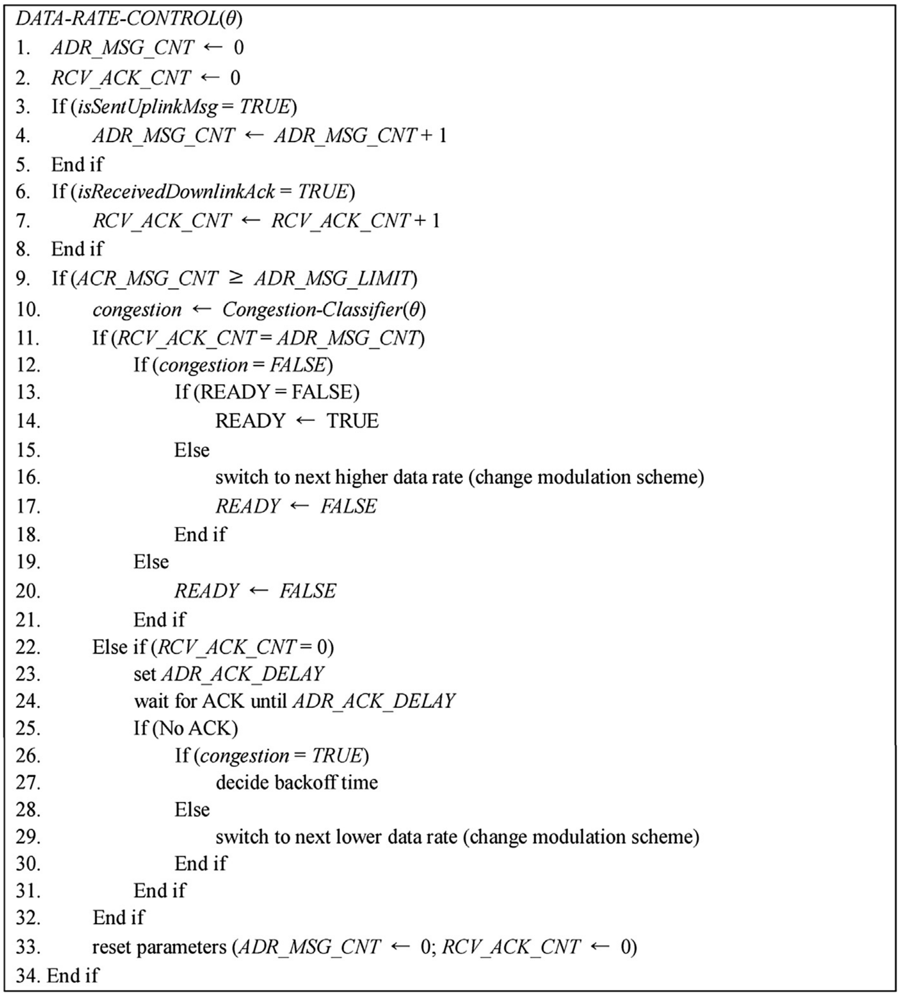
The weight (θ_j) of the j^{th} data attribute (x_j) is updated by the result ($y^{(i)}$) and the prediction ($h(x^{(i)})$) of the i^{th} training example. Using the weights updated by learning, the hypothesis for congestion estimation becomes more accurate. In the next estimation, the computation result of the hypothesis is used and the data rate controller in Fig. 4 performs different operations according to the prediction.

In the proposed method, the congestion classifier has two types. The congestion classifier in the network server side collects training examples from end-devices and performs learning to obtain the best weights. Then, the weights are periodically updated from the network server to gateways. The gateways broadcast the updated weights to the end-devices, and the end-devices can judge congestion using the broadcasted θ. Because the network server manages end-devices, it can gather numerous training examples and has sufficient computing resources. Thus, the network server learns to find the best θ and shares the result of the learning. End-devices use the shared θ to classify the congestion status (Fig. 5).

3.2. Data rate controller

The data rate controller of the proposed method aims to avoid an unnecessary change of a modulation scheme to drop the data rate. Because most IoT traffic consists of short messages, switching a modulation scheme during congestion is inappropriate. Instead of switching the modulation scheme in order to drop the data rate and extend network coverage, adjusting the backoff time is needed to avoid congestion. Therefore, the proposed data rate controller exploits the result of the congestion classifier. According to the result, it determines whether to switch to a lower data rate or adjust the backoff time. Fig. 6 represents the modified adaptive data rate control method for the LoRaWAN.

The proposed data rate control method employs two counters: *ADR_MSG_CNT* and *RCV_ACK_CNT*. *ADR_MSG_CNT* adds one when the end-device sends an uplink message, and *RCV_ACK_CNT* adds one when the end-device receives a downlink ACK message (*line* 3–8). When *ADR_MSG_CNT* reaches *ADR_MSG_LIMIT*, the end-device checks the congestion status using *Congestion-Classifier* and the shared θ from its gateway (*line* 9–10). The *Congestion-Classifier* in the end-device performs the calculation of the hypothesis with the recent weight θ, and it judges the congestion status by the result of the calculation.



```
DATA-RATE-CONTROL($\theta$)
1. ADR_MSG_CNT ← 0
2. RCV_ACK_CNT ← 0
3. If (isSentUplinkMsg = TRUE)
4.     ADR_MSG_CNT ← ADR_MSG_CNT + 1
5. End if
6. If (isReceivedDownlinkAck = TRUE)
7.     RCV_ACK_CNT ← RCV_ACK_CNT + 1
8. End if
9. If (ACR_MSG_CNT ≥ ADR_MSG_LIMIT)
10.    congestion ← Congestion-Classifier($\theta$)
11.    If (RCV_ACK_CNT = ADR_MSG_CNT)
12.        If (congestion = FALSE)
13.            If (READY = FALSE)
14.                READY ← TRUE
15.            Else
16.                switch to next higher data rate (change modulation scheme)
17.                READY ← FALSE
18.            End if
19.        Else
20.            READY ← FALSE
21.        End if
22.    Else if (RCV_ACK_CNT = 0)
23.        set ADR_ACK_DELAY
24.        wait for ACK until ADR_ACK_DELAY
25.        If (No ACK)
26.            If (congestion = TRUE)
27.                decide backoff time
28.            Else
29.                switch to next lower data rate (change modulation scheme)
30.            End if
31.        End if
32.    End if
33.    reset parameters (ADR_MSG_CNT ← 0; RCV_ACK_CNT ← 0)
34. End if
```

Fig. 6. Modified data rate control.

After obtaining the congestion status, the end-device compares *ADR_MSG_CNT* to *RCV_ACK_CNT*. If *ADR_MSG_CNT* has the same value, it means all uplink messages were successfully sent. When congestion does not occur, if all uplink messages are successfully sent within the given time (i.e., twice the *ADR_MSG_LIMIT*), the end-device tries to switch to the next higher data rate by changing the modulation scheme (*line* 11–21). If the end-device does not receive any downlink ACK messages for the *ADR_MSG_LIMIT*, the end-device sets *ADR_ACK_DELAY* and waits for the downlink ACK message. When there is no reception of the downlink message (i.e., ACK message), the end-device checks for network congestion. If congestion occurs, the end-device chooses a long backoff time for random backoff. Otherwise, it switches to the next lower data rate to extend the network coverage (line 22–31).

In the LoRaWAN, adaptive data rate control is performed to adjust the network coverage. The end-devices can control the data rate and the network coverage by changing the modulation scheme of the RF module. The modulation scheme for a low data rate leads to long-range communication. In general, the end-devices switch the data rate according to the reception of the downlink ACK messages. However, changing the modulation scheme without considering network congestion can cause inefficient data transmission. Thus, in the proposed method, the congestion status is estimated by learning and it is applied to determine data rate control in an adaptive manner. Using the proposed method, data transmission can be improved in the LoRaWAN.

4. Performance analysis

4.1. Network model

For analysis, the network can be modeled by a Continuous Time Markov Chain (CTMC). It is assumed that the network has three states such: idle state (state 0), switching state (state 1), and switching state with congestion (state 2). In the idle state, the network has good conditions to transmit data over the LoRaWAN. However, in the switching state, the network experiences poor LoRAWAN conditions. Thus, LoRaWAN end-devices try to control their data rate to extend their coverage. As mentioned previously, the proposed scheme performs different actions in the switching state with congestion, as congestion is not a problem of network coverage.

The network model is represented in Fig. 7. The idle sate is changed to the switching state, the switching state is changed to the switching state with congestion, and the switching state with congestion is changed to the idle state. The state change is repeated. With an exponential arrival rate λ, the network enters the switching state. The switching state has μ_1 as an exponential service rate. That is, the network stays in the switching state for $1/\mu_1$ time and then enters the switching state with congestion. In the switching state with congestion, the network has μ_2 as an exponential service rate and stays for $1/\mu_2$ time. Because the wireless status of the network in each state is different, the end-devices of the LoRaWAN perform the given role according to the state.

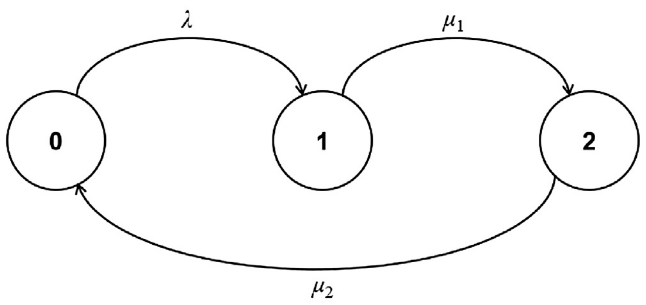


Fig. 7. The network model.

The probability in CTMC can converge to a limiting value that is independent of the initial state. To obtain the stationary probabilities of each state in the network model, the limiting probability [25–27] is applied. That is, by the limiting probability, the probability of leaving state *i* is the same as the probability of entering state *i*. Thus, the network model in Fig. 7 is represented as follows:

$$\begin{array}{cc} State & prob.ofleaving = prob.ofentering \\ 0 & P_0\lambda = P_2\mu_2 \\ 1 & P_1\mu_1 = P_0\lambda \\ 2 & P_2\mu_2 = P_1\mu_1 \end{array} \tag{13}$$

In addition, the summation of the probability of each state is as follows:

$$\sum_i P_i = 1. \tag{14}$$

Then, through Eqs. (13) and (14), the probability of each state of the network model can be represented as follows:

$$\begin{aligned} P_1 &= \frac{\lambda}{\mu_1}P_0 \\ P_2 &= \frac{\lambda}{\mu_2}P_0 \\ P_0 &= P_0 \\ 1 &= \left(1+\frac{\lambda}{\mu_1}+\frac{\lambda}{\mu_2}\right)P_0 \end{aligned} \tag{15}$$

The stationary probabilities of the states of the network model are obtained as follows:

$$\begin{aligned} P_0 &= \frac{\mu_1\mu_2}{\mu_1\mu_2+\lambda\mu_2+\lambda\mu_1} \\ P_1 &= \frac{\lambda\mu_2}{\mu_1\mu_2+\lambda\mu_2+\lambda\mu_1} \\ P_2 &= \frac{\lambda\mu_1}{\mu_1\mu_2+\lambda\mu_2+\lambda\mu_1} \end{aligned} \tag{16}$$

4.2. Transmission delay

When the LoRaWAN is changed according to the network model in Fig. 7, the wireless condition is also changed. The wireless link error is considered in the switching state (state 1) and the congestion error is considered in the switching state with congestion (state 2). In the switching state, the end-devices switch to a lower data rate. The wireless link error is assumed to be 0.03, and the congestion error is assumed to be 0.05. When the packet loss at link *j* is represented as p_j, the packet delivery rate (p_d) can be

$$p_d = 1 - p_j. \tag{17}$$

Table 1
Data rate configurations in LoRaWAN.

Data rate level	Bandwidth	Data rate (bps)
DR0	125 kHz	980
DR1	125 kHz	1760
DR2	125 kHz	3125
DR3	125 kHz	5470

Among *N* packets, the number of successfully transmitted packets can be calculated with the probability of successful transmission in the network. It is represented as follows:

$$\Phi(N) = \sum_{m=1}^{N} m \cdot P_{tx}(N, m). \tag{18}$$

The probability of successful transmission is obtained as follows:

$$P_{tx}(N, m) = \binom{N}{m} \cdot P_d^m \cdot (1 - P_d)^{N-m}. \tag{19}$$

Through the number of successfully transmitted packets, the transmission delay can be computed. The proposed scheme modifies the operation of the end-device as shown in Fig. 6 and the operation is different from the existing LoRa scheme in state 2. Thus, to compare the performance of the proposed scheme and the existing LoRa scheme, transmission delay in state 2 is considered.

The proposed scheme deals with LoRaWAN class A and uses the US 902–928 MHz ISM frequency band. The LoRaWAN in the ISM frequency band has several RF configurations. The proposed scheme assumes that wireless communication exploits the 125 kHz bandwidth. Then, four data rate levels are used to adjust the data rate of an RF module in LoRaWAN devices: DR0, DR1, DR2, and DR3. Each level has data rates as shown in Table 1.

The transmission delay in state 2 can be computed with the number of successfully transmitted packets and data rate during transmission. It is represented as follows:

$$T = \frac{\Phi(N) \times 100 \times 8}{r} + T_{BO}, \tag{20}$$

where *N* is the number of sent packets in the end-device and $\Phi(N)$ is the number of successfully transmitted packets in Eq. (18). *r* is the data rate of the end-device during data transmission. T_{BO} is the backoff time for $\Phi(N)$ and is adjusted in state 2 in the proposed scheme. For the calculation of the transmission delay, the packet size is set to 100 bytes and packet loss in congestion is assumed as 5%. In addition, when congestion occurs, the proposed scheme assumes that the backoff time is determined between 2 and 6 s. In the congestion condition, the proposed scheme maintains the data rate, but the existing scheme switches to the next lower data rate. Thus, if the initial data rate level is DR3, the existing scheme switches to DR2. Fig. 8–10 show the average transmission delay according to the number of sent packets. When the number of sent packets is 30, 40, and 50, the number of successfully transmitted packets at the network server becomes 28, 37, and 47, respectively.

The average transmission delay depends on the number of sent packets. When the number of sent packets is increased, the transmission delay is also increased because the duration for total data traffic delivery is increased. The initial data rate of the RF module in the end-devices also affects the transmission delay. The existing scheme on adaptive data rate control drops the data rate to extend network coverage. Thus, the duration of the existing scheme grows with a low data rate, and it leads to a long transmission delay. In an IoT network, a lot of data traffic is generated and delivered to a network. For robust network connectivity, adaptive data rate control by adjusting the RF modulation configuration is necessary, but

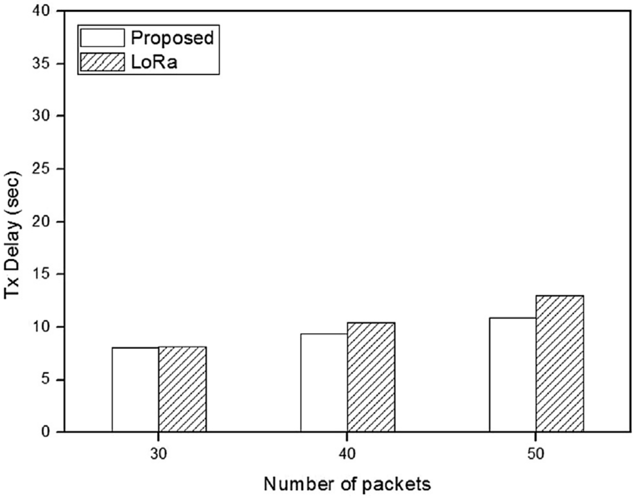


Fig. 8. Initial data rate level is DR3.

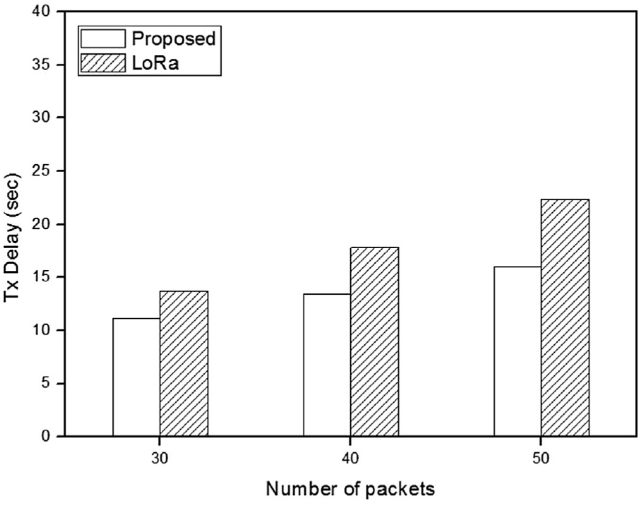


Fig. 9. Initial data rate level is DR2.

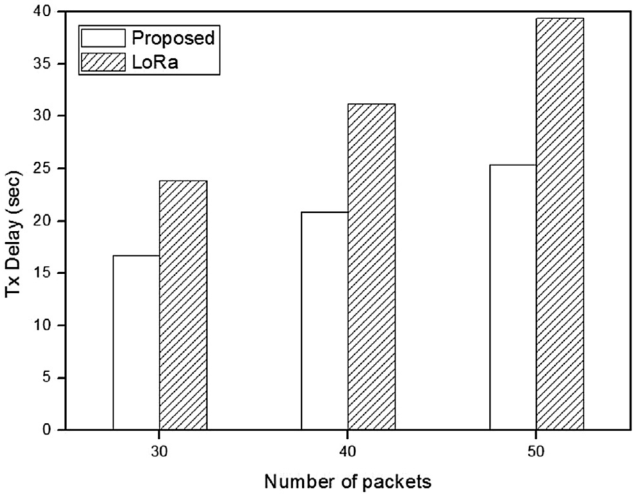


Fig. 10. Initial data rate level is DR1.

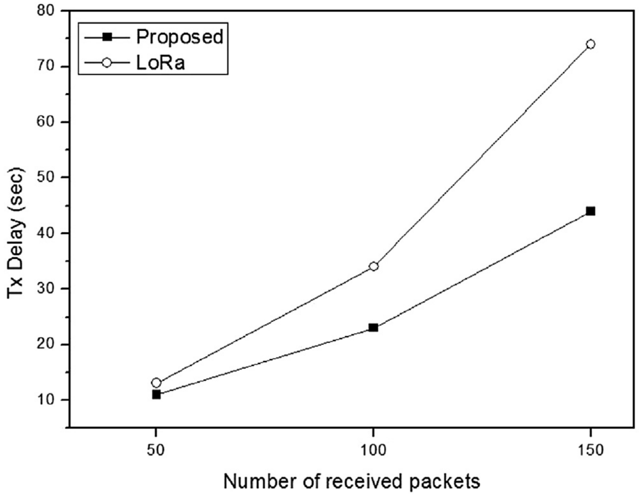


Fig. 11. Transmission delay according to the number of received packets.

as mentioned earlier, congestion is not a problem for network connectivity. Thus, classifying the problem is required for efficient data transmission in LPWAN-based IoT networking.

Fig. 11 shows the transmission delay according to the number of received packets at the network server. In Fig. 11, the network server receives 150 packets and the end-device changes the data rate after every 50 packets. The initial data rate is set to DR3, and it is dropped sequentially every 50 packets. As shown in the figure, the proposed scheme shows better delay performance than the existing scheme. The existing LoRaWAN depends on the reception of ACK messages to control the data rate regardless of network congestion. It drops the data rate by changing the RF modulation in the congestion condition. Thus, the low data rate causes a long transmission delay when congestion occurs. However, the proposed scheme maintains the data rate and increases the backoff time when congestion occurs. It learns the network status, such as congestion using network parameters (e.g., data rate, received signal strength, number of connections at the gateway), and the learning results are applied to data rate control. Thus, although the proposed scheme has a longer backoff time than the existing scheme, it shows less transmission delay and can provide efficient data transmission in a congestion environment.

5. Conclusion

Recently, LPWAN researchers have focused on intelligent services in wide areas such as smart city, smart factory, smart agriculture, etc. The LPWAN has become an important communication technology to implement the IoT networking and CPS of wide areas services. The LoRaWAN, the representative communication technology of the LPWAN, controls the data rate of an end-device to maintain robust network connectivity. It determines the network condition according to the reception status of ACK messages. If the ACK message reception fails, the LoRaWAN decreases the data rate. This may lead to inefficiency in data transmission, because network congestion is not a network connectivity problem. For efficient data transmission in the LoRaWAN, it is required to control the data rate accurately considering the network status. That is, when a connectivity problem occurs, the data rate should be adjusted.

Therefore, the proposed method tries to determine the network status, such as congestion. It learns using network parameters, such as transmission rate, received signal strength, and number of data sessions. For learning, the proposed method employs a logistic regression algorithm. When congestion is predicted, the proposed scheme adjusts the backoff time instead of dropping the data rate.

The learning mechanism used to classify the congestion requires extensive computing resources. In the proposed scheme, because every network data packet is transmitted to a network server, the network server performs learning. The result of the learning is delivered to the end-devices and is used to predict congestion. The proposed scheme shows better performance by reflecting congestion in adaptive data rate control, but a large portion of the computing for learning is operated in a centralized machine. Therefore, the distributed learning of end-devices needs to be studied in future work.

Acknowledgements

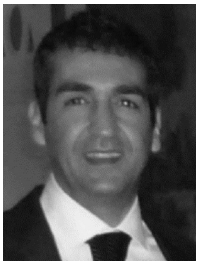
This research was supported by the Soonchunhyang University Research Fund (No. 20160220). This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2016R1A2B4011069).

References

[1] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions, Elsevier Future Gener. Comput. Syst. 29 (7) (2013) 1645–1660.
[2] L. Atzori, A. Iera, G. Morabito, The internet of things: a survey, Elsevier Comput. Netw. 54 (15) (2010) 2787–2805.
[3] M. Chen, J. Wan, F. Li, Machine-to-machine communications: architectures, standards and applications, KSII Trans. Internet Inf. Syst. 6 (2) (2012) 480–497.
[4] D. Bandyopadhyay, J. Sen, Internet of Things: applications and challenges in technology and standardization, Wireless Pers. Commun. 55 (1) (2011) 49–69.
[5] S. Kim, W. Na, Safe data transmission architecture based on cloud for Internet of Things, Wireless Pers. Commun. 86 (1) (2016) 287–300.
[6] C.-Y. Lin, S. Zeadally, T.-S. Chen, C.-Y. Chang, Enabling cyber physical systems with wireless sensor networking technologies, Int. J. Distrib. Sens. Netw. 2012 (2012) (ID 489794).
[7] V. Gunes, S. Peter, T. Givargis, F. Vahid, A Survey on concepts, applications, and challenges in cyber-physical systems, KSII Trans. Internet Inf. Syst. 8 (12) (2014) 4242–4267.
[8] D. Culler, D. Estrin, M. Srivastava, Guest editors introduction: overview of sensor networks, IEEE Comput. 37 (8) (2004) 41–49.
[9] D.-Y. Kim, J. Cho, B.-S. Jeong, Practical data transmission in cluster-based sensor networks, KSII Trans. Internet Inf. Syst. 4 (3) (2010) 224–242.
[10] D.-Y. Kim, Z. Jin, J. Choi, B. Lee, J. Cho, Transmission power control with the guaranteed communication reliability in WSN, Int. J. Distrib. Sens. Netw. 2015 (2015) (ID 632590).
[11] X. Xiong, K. Zheng, R. Xu, W. Xiang, P. Chatzimisios, Low power wide area machine-to-machine networks: key techniques and prototype, IEEE Commun. Mag. 53 (9) (2015) 64–71.
[12] A. Augustin, J. Yi, T. Clausen, W.M. Townsley, A study of LoRa: long range & low power networks for the Internet of Things, MDPI Sens. 16 (9) (2016) (ID 1466).
[13] M. Centenaro, L. Vangelista, A. Zanella, M. Zorzi, Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios, IEEE Wireless Commun. 23 (5) (2016) 60–67.
[14] D.-Y. Kim, M. Jung, Data transmission and network architecture in long range low power sensor networks for IoT, Wireless Pers. Commun. 93 (1) (2017) 119–129.
[15] LoRa Alliance, https://www.lora-alliance.org, 2016.
[16] SIGFOX, http://www.sigfox.com, 2016.
[17] Weightless, http://www.weightless.org, 2015.
[18] N. Sornin, M. Luis, T. Eirich, T. Kramp, O. Hersent, LoRa Alliance LoRaWAN specification, LoRaWAN Specification, Release v1.0, 2015.
[19] IEEE, IEEE 802.15.4 standard, IEEE 802.15. 4-2006 Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPAN), 2006.
[20] J.A. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, B. Heile, IEEE 802.15.4: A developing standard for low-power low-cost wireless personal area networks, IEEE Netw. 15 (5) (2001) 12–19.
[21] S. Marsland, Machine, Learning an Algorithmic Perspective, Chapman & Hall, New York, NY, USA, 2009.
[22] E.F. Flushing, J. Nagi, G.A.D. Caro, A mobility-assisted protocol for supervised learning of link quality estimates in wireless networks, Proc. of International Conference on Computing, Networking and Communications (ICNC) (2012) 137–152.
[23] M.A. Alsheikh, S. Lin, D. Niyato, H.-P. Tan, Machine learning in wireless sensor networks: algorithms, strategies, and applications, IEEE Commun. Surv. Tutorials 16 (4) (2014) 1996–2018.
[24] M. Di, E.M. Joo, A survey of machine learning in wireless sensor networks form networking and application perspectives, Proc of International Conference on Information, Communications & Signal Processing (2007).
[25] S.M. Ross, Probability Models for Computer Science, Harcourt/Academic Press, 2001.
[26] R.D. Yates, D.J. Goodman, Probability and Stochastic Processes: a Friendly Introduction for Electrical and Computer Engineers, Wiley, 2004.
[27] K.S. Trivedi, Probability and Statistics with Reliability, Queuing and Computer Science Applications, Wiley, 2001.

Dae-Young Kim received his B.E. degree in electronics engineering, M.S. and Ph.D. degrees in computer engineering from Kyung Hee University, Korea in 2004, 2006, and 2010, respectively. From 2010–2013, he was a research staff at Communication R&D Lab. of LIG Nex1, Korea. From 2013–2015, he was a research staff at Network Team of AirPlug, Inc., Korea. Currently, he is an assistant professor of department of software engineering at Changshin University, Korea.H is research interests include mobile networking & computing, wireless sensor networks and embedded system softwares for IoT.

Seokhoon Kim received the B.E. and Ph.D. degrees in computer engineering from Kyunghee University, Korea, in 2000 and 2004, respectively. He was an Assistant Professor in the Department of Mobile Communications Engineering at Changshin University, Changwon, Korea. He currently works as an Assistant Professor in the Department of Software Engineering at Soonchunhyang University, Asan, Korea. His research interests comprise Cloud Computing, Internet of Things, Software Defined Networking, Mobile System/Communications, and Machine Learning based on Bigdata.



Houcine Hassan is associate Professor of the Department of Computer Engineering in the Polytechnic University of Valencia, Spain, since 1994. He received a MS and PhD. degree in Computer Engineering from the Polytechnic University of Valencia, in 1993 and 2001 respectively. He joined the Industrial Informatics group in 1993, where he is participating in several research projects. His research interests covers a number of aspects of the development of hardware and software architectures including real-time systems support, embedded systems, hardware/software co-design, ubiquitous computing, AI agent based systems, autonomous systems, sensor networks, robotic architectures, behaviour and emotional systems, application and scheduling integration, QoS.

Dr. James J. (Jong Hyuk) Park received Ph.D. degrees in Graduate School of Information Security from Korea University, Korea and Graduate School of Human Sciences from Waseda University, Japan. From December 2002 to July 2007, Dr. Park had been a research scientist of R&D Institute, Hanwha S&C Co., Ltd., Korea. From September 2007 to August 2009, He had been a professor at the Department of Computer Science and Engineering, Kyungnam University, Korea. He is now a professor at the Department of Computer Science and Engineering and Deptartment of Interdisciplinary Bio IT Materials, Seoul National University of Science and Technology (SeoulTech), Korea. Dr. Park has published about 200 research papers in international journals and conferences. He has been serving as chairs, program committee, or organizing committee chair for many international conferences and workshops. He is a founding steering chair of some international conferences –MUE, FutureTech, CSA, UCAWSN, etc. He is editor-in-chief of Human-centric Computing and Information Sciences(HCIS) by Springer, International Journal of Information Technology, Communications and Convergence (IJITCC) by InderScience, and Journal of Convergence (JoC) by KIPS CSWRG. He is Associate Editor/Editor of 14 international journals including 8 journals indexed by SCI(E). In addition, he has been serving as a Guest Editor for international journals by some publishers: Springer, Elsevier, John Wiley, Oxford Univ. press, Hindawi, Emerald, Inderscience. His research interests include security and digital forensics, Human-centric ubiquitous computing, context awareness, multimedia services, etc. He got the best paper awards from ISA-08 and ITCS-11 conferences and the outstanding leadership awards from IEEE HPCC-09, ICA3PP-10, IEE ISPA-11, and PDCAT-11. Furthermore, he got the outstanding research awards from the SeoulTech, 2014. Dr. Park' s research interests include Human-centric Ubiquitous Computing, Vehicular Cloud Computing, Information Security, Digital Forensics, Secure Communications, Multimedia Computing, etc. He is a member of the IEEE, IEEE Computer Society, KIPS, and KMMS.