Spectrum Sharing for Internet of Things System in Periodic Transmission

Mai Ohta, Masaki Amano, and Makoto Taromaru
Fukuoka University, Fukuoka, Japan

***Abstract*—This paper focuses on future Internet of things (IoT) devices equipped with a low-cost oscillator. An IoT system consists of many wireless devices and a base station (gateway) and is designed to collect multiple pieces of useful information. Each IoT device transmits signals periodically at regular intervals and is equipped with a low-cost oscillator. In previous work, using computer simulation, we have shown that collisions occur between packets transmitted by the IoT devices, as a result of the poor frequency stability caused by the cheap oscillator. Moreover, we have previously proposed an algorithm that enables an IoT device to modify its communication cycle: a regular packet transmission is delayed by predicting the interval between received packets at a gateway. However, packets from the device whose transmission timing is modified collide with the following packets from other IoT devices, and the total number of modifications is greatly increased by this algorithm. The work reported in this paper aims to suppress packet collisions more effectively than the previous algorithm. The performance of the new algorithm is confirmed by simulation.**

I. Introduction

The number of devices equipped with wireless communication functions has been increasing because of the Internet of things (IoT). In an IoT system, a base station collects sensor information from many nodes. However, the wireless devices for IoT do not include precise oscillators because of their high cost. The cost of each node has to be reduced because many nodes are used in an IoT system. In addition, many IoT nodes need to share a limited spectrum for their wireless communications.

The use of cheap oscillators causes collision of packets between nodes even if the transmission time of each node is arranged to avoid collision before the start of the IoT service. When the IoT system transmits sensor information at regular intervals, the transmission time becomes incorrect as time passes because of the inaccurate oscillator. Therefore, an IoT system requires a collision avoidance method.

In this paper, we assume that the IoT base station predicts a packet collision by measuring the interval between received packets and transmits a control packet to the node whose packets are likely to collide. The corresponding nodes change their next transmission time when they receive the control packet from the base station. Therefore, the nodes can avoid packet collisions, even with cheap oscillators.

II. System Model

In this paper, a simple IoT system is considered. Our system model assumes the existence of only one IoT system, so interference from other systems is not considered. The IoT system consists of many IoT devices, which are wireless nodes, and an IoT gateway; mobility is not considered. All IoT devices can connect only to the gateway; they cannot communicate with the gateway by a multi-hop mechanism.

In the IoT system, the devices observe their surroundings (e.g., temperature, humidity, power consumption, and degree of congestion). The information observed by the IoT devices is transmitted to the gateway. The IoT devices pass information to the gateway at regular intervals when the objective is to monitor the surroundings. This paper focuses on this type of regular and periodic packet transmission.

Conversely, the IoT device may be used as a type of alert for notifying the user about the occurrence of an abnormality or event; in this case, the IoT device transmits a signal if and when such an event occurs. However, this paper does not consider such an event-driven IoT system.

One of the features of IoT systems is that they can use a large number of devices to collect information from many different locations. An IoT system can monitor the surrounding environment by making use of the collected information. Moreover, IoT is expected to be used for collecting many different types of data, and various IoT services are becoming available from different IoT providers. To satisfy this demand, wireless devices for IoT services will be mass-produced at a low price in future.

If several circuits are available for constructing an IoT device, a cheaper circuit will be preferred. This paper focuses on an oscillator produced at a low price. The price of oscillators has a particularly strong effect on their frequency stability, which includes the accuracy of the frequency and its variation over time. The price of oscillators is decided by the type of material (e.g., crystal, silicon, or ceramic) that is used. However, the frequency stability of IoT devices differs even if the same material is used.

In this paper, it is assumed that all IoT devices act with their own frequency stability because of external conditions (e.g., temperature variation and age deterioration). An IoT device cannot recognize the characteristics of its own frequency stability. Therefore, packet collisions occur with the passage of time even if the IoT devices can initially transmit packets without collision. In this paper, we assume that the gateway, which regularly receives packets from IoT devices, informs each IoT device of the detection of collision with packets from other devices.

978-988-14768-8-3/20/$31.00

III. Proposed Algorithms for Collision Avoidance

The IoT devices begin the communication by sending packets to an IoT gateway. The gateway can communicate with the IoT devices within its own network. The packets transmitted by each IoT device include the identification of the transmitting device. When packet collisions are predicted, by observing the reception timings of packets from each IoT device, the gateway commands the corresponding device to modify the timing of its next transmission.

In this section, we explain two algorithms for collision avoidance in a regular communication cycle. The proposed algorithms are presented in the following subsections.

A. Collision Avoidance by Delaying Transmission Timing: DLTT

An existing collision avoidance algorithm [1] works by delaying transmission timing. In summary, the gateway instructs the IoT device to delay its next transmission when a packet collision is predicted, regardless of the time when the following packet is received by the other node. In this manner, the transmission timing of the IoT devices is always delayed whenever the gateway predicts packet collision. The algorithm proceeds as follows:

1) Each IoT device in the system observes its surroundings and transmits data at regular intervals; the transmission interval is T^{Reg}.
2) The IoT gateway receives the transmitted packets and recognizes the time interval between the received packets, T_x, where x is the index of the reception time interval.
3) The time interval is compared to a threshold, γ, at the gateway:
$$T_x < \gamma. \tag{1}$$
When (1) is true, the gateway determines that the timing of the next transmission from the corresponding device should be modified. Otherwise, the procedure is repeated from Step 1.
4) If (1) is true, the gateway recognizes the identification of the IoT device whose packet is predicted to collide with the packet.
5) The gateway transmits a control command, including an acknowledgement (ACK) signal, to the corresponding device.
6) The IoT device that receives the control command, including ACK, delays the timing of its next transmission by a constant time interval, δ^D, in response to the gatewayfs command. The receiving IoT device then transmits the next packet after the modified time interval.
7) The procedure is repeated from Step 1.

This method can avoid collision between packets by delaying the packet transmission timing of the offending device. In this method, the transmissions of the whole IoT network are delayed because the packets transmitted by each IoT device are delayed every time packet collision is predicted. A longer lapse time increases the impact of the time delay. It is possible that this method could prevent the IoT system collecting the required amount of data within a certain period of time.

B. Collision Avoidance by Dynamically Determining Transmission Timing: DDTT

The algorithm described above delays an IoT devicefs transmission timing when packet collision is detected. Moreover, the delay time interval is constant. With this algorithm, the network efficiency is reduced because the transmission timings of the whole IoT network are delayed.

In this subsection, we propose an algorithm for increasing the flexibility of control. The proposed algorithm is as follows:

1) Each IoT device in the system observes its surroundings and transmits data at regular intervals; the transmission interval is T^{Reg}.
2) The IoT gateway receives the transmitted packets and recognizes the time intervals before and after the received packet, which are denoted by T_x and T_{x+1}, respectively.
3) Each time interval is compared to a threshold, γ, at the gateway:
$$T_x < \gamma, \tag{2}$$
$$T_{x+1} < \gamma. \tag{3}$$
When either (2) or (3) is true, the gateway determines that the timing of the next transmission from the corresponding device should be modified. Otherwise, the procedure is repeated from Step 1.
4) If either (2) or (3) is true, the gateway recognizes the identification of the IoT device whose packet is predicted to collide with the packet.
5) At the gateway, a shift time interval, δ^{S}, is calculated using the following equation:
$$\delta^{\mathrm{S}} = \frac{T_{x+1} - T_x}{2}, \tag{4}$$
6) The gateway transmits a control command containing the computed shift time interval, δ^{S}, including an ACK signal, to the corresponding device.
7) The IoT device that receives the control command, including the ACK, modifies the timing of its next transmission. The next transmission time, $T_{n,j}^{\mathrm{NXT}}$, is modified to $T_{n,j}^{\mathrm{MOD}}$ defined as follows:
$$T_{n,j}^{\mathrm{MOD}} = T_{n,j}^{\mathrm{NXT}} + \delta^{\mathrm{S}}, \tag{5}$$
where n is the index of the device and j is the index of the packet transmitted by each device. The receiving IoT device then transmits the next packet after the modified time interval.
8) The procedure is repeated from Step 1.

C. Deciding the Threshold, γ

In the algorithm proposed in this paper, the threshold, γ, is used for predicting whether packets transmitted from more than one node collide. Therefore, our proposed system has to decide γ to be able to predict a packet collision before the

packets collide. It is worth noting that, in the proposed method, the adjustment of transmission timing may be a burden for battery-powered IoT devices.

If the threshold is too short, packet collision occurs before transmission timing is changed. Conversely, if the threshold is too long, the number of changes in transmission timing is excessively large. Therefore, to realize good performance by using the proposed method, the value of the threshold, γ, is important. The threshold has to be decided considering both the time interval and the length of the packet if the packet length changes at every transmission.

IV. Simulation Results

A. Simulation Parameters

This subsection describes the parameters of the simulation. In the simulation, we assumed an IoT system employing LoRaWAN as the low-power wide-area network (LPWAN), with IoT devices uniformly distributed across an area within a radius of 5 km from a gateway.

Each IoT device had several frequency stabilities. A time jitter, caused by poor frequency stability, was set to a random value generated by the standard normal distribution with a mean of 20 ppm. In this simulation, its several time jitters remained unchanged during the simulation.

The next transmission time, $T_{n,j}^{\mathrm{NXT}}$, is determined by the expression:

$$T_{n,j}^{\mathrm{NXT}} = T_{n,j-1}^{\mathrm{NXT}} + T_n^{\mathrm{CYC}} + T_n^{\mathrm{JIT}}, \tag{6}$$

where $T_{n,j-1}^{\mathrm{NXT}}$ is the transmission time of the previous packet, T_n^{CYC} is the transmission time interval of the n-th device, and T_n^{JIT} is the time jitter of the n-th device. If there is no time jitter, the next transmission time, $T_{n,j}^{\mathrm{NXT}}$, is determined by:

$$T_{n,j}^{\mathrm{NXT}} = T_{n,j-1}^{\mathrm{NXT}} + T_n^{\mathrm{CYC}}. \tag{7}$$

In the LoRaWAN, the bit rate is defined by:

$$R_{\mathrm{bit}} = SF \cdot \frac{BW}{2^{SF}}, \tag{8}$$

where BW is the bandwidth and SF is the spreading factor (SF), which is generally an integer between 7 and 12. However, in Japan, an integer between 7 and 10 is used as SF, to comply with the standards of the Association of Radio Industries and Businesses (ARIB) [2], which serves as the standards development organization of Japan. The length of a packet, L_n, is defined by:

$$L_n = \frac{N_{\mathrm{bit}}}{R_{\mathrm{bit}}}, \tag{9}$$

where N_{bit} is the number of bits of a transmission packet.

Table I shows the simulation parameters. In this simulation, IoT nodes performed carrier sense before packet transmissions, following the rule in Japan. However, carrier sense is a technique for avoiding a packet collision just before the packet transmission. In this paper, packet loss caused by packet collisions was judged by the SF and the Signal-to-Interference power Ratio (SIR). The tolerance to the packet collision differs depending on the SF and SIR [3]. Moreover, it was assumed that packets are lost when the number of nodes is three and more.

TABLE I
Simulation Parameters.

System for LPWAN	LoRaWAN
Spreading factor	7 to 12
Transmission power	20 mW
Transmission payload	240 bit
Transmission time interval of each device	10 min
The number of device	1000
Communication area	Radius of 5 km
Location of devices	Uniform distribution
Transmission antenna gain	3 dBi
Antenna height at device	30 m
Frequency stability	20 ppm
The number of gateway	1
Reception antenna gain	3 dBi
Antenna height at gateway	30 m
Frequency band	920 MHz band
Width of street	10 m
Interval between buildings	10 m
Constant time interval, δ^D, for DLTT	0.1 sec

B. Simulation Evaluations

In this paper, it is assumed that frequency stability is caused by the time jitter of an oscillator. In the experiments, the average frequency stability was 20 ppm. Each frequency stability includes a deviation, which was a normally distributed random number.

Fig. 1 shows the transmission success rate for various numbers of nodes (1000, 1200, 1500, and 2000). After dramatically decreasing during the first 25 hours, the transmission success ratio then decreased more gradually in this simulation. Without the proposed method, many packet collisions occurred but the performance stabilized as time passed. That is, a certain number of nodes could communicate with the gateway regardless of whether the proposed method was used. Moreover, as shown in Fig. 1, the transmission success ratio was affected more when the number of nodes was larger. In the remaining simulations, the number of nodes was set to 1000.

Fig. 2 shows the number of modifications (of transmission timing) performed using each of the collision avoidance methods. In Fig. 2, the effects depend on the threshold, γ, which is used for deciding when the transmission timing of a node should be modified. The results of this simulation and the next were obtained for threshold value of 0.01, 0.1, and 0.2 s. In the case of $\gamma = 0.01$, the number of modifications was the smallest of all results, because this means that the transmission timing is unchanged unless the interval between the received packets is less than 0.01 s. Therefore, the number of modifications increases as the threshold, γ, increases. If the threshold is too large, the number of controls performed by a node increases because of the modifications of transmission timing.

Fig. 3 shows the results of bit rate when each of the proposed methods was used, and when collision avoidance was not performed. The bit rate performance improved, compared with the baseline method, when the DLTT method was used

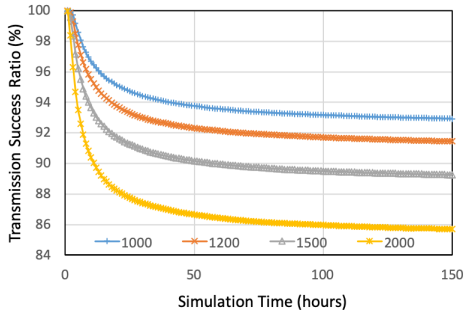

Fig. 1. Transmission success ratio for various numbers of nodes.

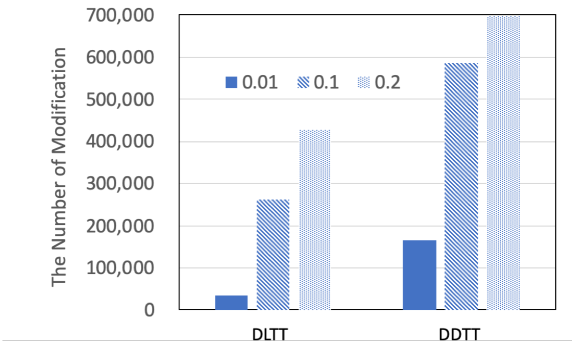

Fig. 2. Number of modifications performed by each algorithm.

with $\gamma = 0.2$ s and when DDTT was used with $\gamma = 0.01$ to 0.2 s. When the threshold, γ, in DLTT was 0.1 s, the bit rate decreased because the threshold was not suitable in this simulation environment. Therefore, the proposed DDTT method was able to improve the bit rate.

These results show that there is a performance trade-off between the number of modifications and the bit rate. Thus, a system using the proposed collision avoidance algorithm has to consider which aspect of performance is most important. Moreover, parameters such as the threshold for modification must be decided by a desired indicator of the system (e.g., communication success ratio, bit rate, or both).

V. Conclusions

This paper focused on the expected widespread use of IoT systems in the LPWAN area. In the IoT system, frequency deviation caused by a poor oscillator leads to packet collision in the transmission cycle. The paper proposed a new collision avoidance method, DDTT, for IoT systems that transmit packets at regular intervals and consist of a gateway and many nodes. In the new method, the transmission timing is dynamically decided by the intervals before and after each received packet. The simulation results show the performance of the number of modifications of transmission timing and bit rate with DDTT, DLTT (a previous method), and with no collision avoidance method. The simulation evaluations show

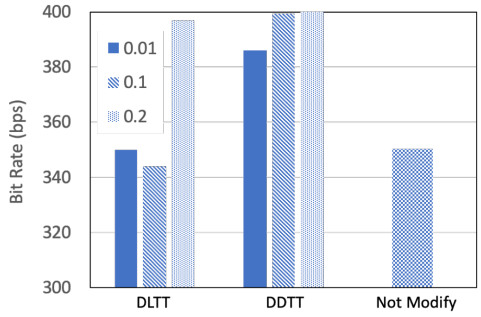

Fig. 3. Bit rate with each algorithm and without collision avoidance algorithm.

that the proposed method can improve the bit rate more than the previous method. However, the IoT system has to use a threshold that is decided according to the degree of priority of the system because the obtained performance depends on this threshold.

Acknowledgment

This research and development work was supported by the MIC/SCOPE #205004001.

References

[1] M. Amano, M. Ohta, and M. Taromaru, "Transmission Timing Control Method for IoT Devices Equipped with Poor Oscillator," in *Proc. 10th Int. Conf. on Ubiquitous and Future Netw. (ICUFN)*, pp. 132-136, July 2019.

[2] 920MHz-Band Telemeter, Telecontrol and Data Transmission Radio Equipment, ARIB STD-T108 Version 1.3, http://www.arib.or.jp/english/html/overview/doc/5-STD-T108v1_3-E1.pdf.

[3] D. Croce, M. Gucciardo, and S. Mangione, "Impact of LoRa Imperfect Orthogonality: Analysis of Link-Level Performance," *IEEE Comm. Letters,* vol. 22, issue 4, pp. 796-799, April 2018.