Master's Thesis
석사 학위논문

Adaptive Tuning of IEEE 802.11 MAC Access Delay for Improving Control Performance in Cyber-Physical Systems

Kyoungbok Kim (김 경 복 金 慶 福)

Department of Information and Communication Engineering

정보통신융합공학전공

DGIST

2016

Master's Thesis
석사 학위논문

Adaptive Tuning of IEEE 802.11 MAC Access Delay for Improving Control Performance in Cyber-Physical Systems

Kyoungbok Kim (김 경 복 金 慶 福)

Department of Information and Communication Engineering
정보통신융합공학전공

DGIST

2016

Adaptive Tuning of IEEE 802.11 MAC Access Delay for Improving Control Performance in Cyber-Physical Systems

Advisor : Professor Kyung-Joon Park

Co-advisor : Professor Jonghyun Kim

by

Kyoungbok Kim
Department of Information and Communication Engineering
DGIST

A thesis submitted to the faculty of DGIST in partial fulfillment of the requirements for the degree of Master of Science in the Department of Information and Communication Engineering. The study was conducted in accordance with Code of Research Ethics[1].

Nov. 13. 2015

Approved by

Professor Kyung-Joon Park (Signature)
(Advisor)

Professor Jonghyun Kim (Signature)
(Co-Advisor)

[1] Declaration of Ethical Conduct in Research: we, as a graduate student of DGIST, hereby declare that we have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. We affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

Adaptive Tuning of IEEE 802.11 MAC Access Delay for Improving Control Performance in Cyber-Physical Systems

Kyoungbok Kim

Accepted in partial fulfillment of the requirements for the degree of Master of Science

Nov. 13. 2015

Head of Committee _______________(인)
Prof. Kyung-Joon Park

Committee Member _______________(인)
Prof. Jonghyun Kim

Committee Member _______________(인)
Prof. Jihwan Choi

MS/IC
201422003
김 경 복. Kyoungbok Kim. The study of Adaptive Tuning of IEEE 802.11 MAC Access Delay for Improving Control Performance in Cyber-Physical Systems. Department of Information and Communication Engineering. 2016. 36p. Advisors Prof. Kyung-Joon Park, Co-Advisors Prof. Jonghyun Kim.

ABSTRACT

In this paper, we study the problem of tuning medium access control (MAC) parameters in IEEE 802.11 to improve control performance in cyber-physical systems (CPS). Our main idea is to propose an algorithm that guarantees control performance and to adjust MAC parameters that affect the delay distribution. In addition, we exploit the fact that control performance depends more on delay distribution than just on the average delay. In legacy networked control systems (NCS) design, the network delay is mostly considered as a random delay distribution, whereas the real network delay has different distribution features like the long-tailed characteristic in IEEE 802.11, so the system model cannot be accurate in the real world.

We make two contributions to this field of research: i) Through the analysis of the IEEE 802.11 MAC access delay model, we control the MAC parameters by reducing the delay variance, that can be critical to control performance as well as the average delay. We propose a MAC controller that controls the MAC parameters with our algorithm. ii) In order to validate our approach, we implement a CPS testbed, in which a remote controller controls a drone via an IEEE 802.11 network. A CPS testbed based simulation is also completed with experiments. We examine the absolute average tracking error of the controller for a given reference. Our results show that the proposed approach can significantly improve control performance.

Keywords: Cyber-physical systems, MAC access delay, IEEE 802.11, control performance

Contents

List of tables

List of figures

I. INTRODUCTION

Recently, the convergence of cyber and physical systems has emerged as a promising field that has transformed conventional embedded systems into so-called cyber-physical systems (CPSs). CPSs are mainly characterized by close integration and coordination among computational and physical processes by means of networking [1], [2], [22].

As a CPS generally controls physical systems through a network, network characteristics in the CPS strongly affect the control performance. Traditionally, networked control systems (NCSs) has been studied for this problem in the automatic control community [3], [23]. However, the main focus in the NCS is mostly on the stability of physical systems, while the network is typically modeled as a source of packet delay and packet loss [18].

In this paper, we study the problem of parameter tuning in IEEE 802.11 MAC for improving control performance in a CPS. Although there have been many studies on tuning of network parameters for the CPS, most existing studies have focused on the average delay characteristics of the network with regard to system stability. Our main idea is to exploit the fact that control performance in a CPS depends more on delay distribution than on just the average delay of delay. We make two contributions: i) By analyzing the IEEE 802.11 delay model, we control the delay variance, which can be more critical to control performance, as well as the average network delay. We propose a medium access control (MAC) controller that controls MAC parameters with our algorithm. ii) In order to validate our approach, we implement a CPS testbed, in which a remote controller controls a drone via an IEEE 802.11 network. A CPS testbed based simulation is also conducted with experiments.

In the network side, the network parameters of the IEEE 802.11 protocol are typically tuned for minimizing the average delay. However, this solution does not necessarily guarantee the best control performance since control performance relies on delay distribution.

For example, a network with a larger average delay and a smaller delay variance may give a better control performance than one with a smaller average delay and a larger delay variance.

First, we use an exact saturated IEEE 802.11 DCF MAC delay model to apply the simpler model which has similar characteristics to the CPS testbed in [4], although an unsaturated model of the IEEE 802.11 DCF model exists in [20]. As a function of MAC parameters, such as the initial contention window size, the number of backoff stages, and the maximum number of retransmissions, we obtain the distribution of medium access control (MAC) access delay. Through the analysis of the delay model, we examine the exact effect on the delay distribution as MAC parameters change. Then, we tune these MAC parameters in order to optimize both the average delay and the delay variance of MAC access delay.

Second, in order to verify our approach, we consider a CPS testbed equipped with camera localization capabilities, in which a remote controller controls a drone via an IEEE 802.11 network. With our CPS testbed based simulation and experiments, we evaluate the control performance of our CPS testbed with IEEE 802.11 MAC access delay when the delay variance decreases. We examine the tracking error of the controller for a given reference. Our preliminary results show that the delay variance can affect the control performance.

Third, we propose the MAC controller with a new algorithm to achieve optimal control performance by tuning MAC parameters. We also verify our algorithm through CPS testbed based simulations. Our results show that the proposed approach can significantly improve control performance.

The rest of this paper is organized as follows: In Section II, we summarize existing work with respect to tuning MAC in CPS. Section III introduces basic background knowledge needed to understand our approach. In Section IV, we formulate the optimization problem from the analysis of the system and delay model. Section V introduces our MAC controller architecture for applying our approach. The simulation and experimental evaluations are presented in Section VI. Finally, Section VII concludes this paper.

II. RELATED WORK

Generally, IEEE 802.15.4 protocol has characteristics similar to IEEE 802.11 because both protocols use a CSMA/CA mechanism, and the characteristic of delay distribution between both protocols has a heavy tailed distribution. In this section, we consider the previous studies about IEEE 802.15.4 MAC Wireless Personal Area Networks (WPAN) protocol as well as IEEE 802.11 in the NCS.

2.1. Tuning MAC of IEEE 802.15.4 for Networked Control Systems (NCSs)

In [8], the change of MAC parameters in IEEE 802.15.4 affected the packet delay and control performance in the NCS. They analyzed the probabilistic MAC access delay model of IEEE 802.15.4 to validate the probability distribution. They also showed the effect of MAC access delay in the NCS through an example in which the effect of MAC access delay could make the control system unstable as MAC parameters change. Thus, the selection of wireless network MAC parameters is one of the important issues for the stability of a control system.

In [9], they proposed the constrained optimization problem of IEEE 802.15.4 MAC to minimize the energy consumption considering the guarantee of the packet transmission reliability, constrained average delay. By modeling the IEEE 802.15.4 MAC for reliable and timely system, they could adjust an adaptive MAC algorithm for satisfying the optimal conditions. In addition, the performance of the MAC algorithm was evaluated by numerical results through Monte Carlo simulations.

2.2. Tuning MAC of IEEE 802.11 for NCS

In [5], they showed the effect of the number of maximum retransmissions which is one of the MAC parameters in NCS performance. They also checked the limit for the optimal number of retransmissions through absolute average tracking error. To achieve optimal control performance, they presented an adaptive strategy to determine the optimum number of retransmissions with a MAC layer controller which dynamically tunes the number of maximum retransmission in a NCS. Simulation results confirmed a MAC controller that can increase the NCS control performance.

III. BACKGROUND

3.1 IEEE 802.11 Protocol

IEEE 802.11 is a standard protocol of wireless local area network (WLAN) communication, which is also called Wi-Fi and is widely used for wireless communication. The standard protocol was released in 1997 and has been developed in various versions of the standard, such as 802.11a, 802.11b, 802.11g, and so on. IEEE 802.11 includes both medium access control (MAC) and physical layer (PHY) specifications in 2.4 GHz ISM, 5 GHz, and 60 GHz bands. In our study, we concentrate on the MAC layer for control performance.

3.2 IEEE 802.11 Medium Access Control (MAC) Protocol

IEEE 802.11 MAC protocol coordinates packet collisions between multiple nodes to share the channel [19]. There are two access methods in IEEE 802.11 MAC. One is the distributed coordination function (DCF) and the other is the point coordination function (PCF). In the DCF, each node contends to use the channel, whereas the PCF is a centralized contention free access method [10]. For various reasons, such as robustness in the real world [11], IEEE 802.11 MAC mainly uses the DCF method. Therefore, we deal with the DCF method in this section.

The DCF employs a carrier sense multiple access with collision avoidance (CSMA/CA) mechanism and a binary exponential backoff algorithm to reduce the collision probability. The operation and MAC parameters of IEEE 802.11 MAC DCF will be introduced to understand the IEEE 802.11 MAC access delay model in the next subsection. When a node wants to transmit packets to another node, the node monitors the channel state during a distributed interframe spacing time (DIFS). If the channel state is 'busy', the node state becomes 'frozen' until the transmission is over. As the node senses the channel state is 'idle', it transmits packets according to a random time. Each node to transmit data has a backoff time counter within the range of the initial contention window (W), which is in the range of [1, W-1]. When the backoff time counter becomes zero, data transmission occurs. The MAC layer knows that the data transmission has been successful by receiving an ACK from a node which received data packets after a short interframe spacing time (SIFS). However, collisions of packets can occur at the channel when some data packets are transmitted at the same time. Then, the binary backoff algorithm works by doubling the range of the contention window, which is equal to the number of backoff stages (m). By doing so, the range of the selected backoff time counter increases and the node has a lower collision probability. The contention window can increase up to the maximum contention window (K) by doubling the range whenever continuous collisions occur. Fig. 3.1 shows how the CSMA/CA mechanism works in the collision situation. After the first and second data transmission finishes successfully, two stations have the same backoff counter equal to 4. Therefore, the next data transmission occurs the collision situation of a packet. When the data transmission is unsuccessful within the ACK timeout, W is doubled and each station has a new backoff time counter.

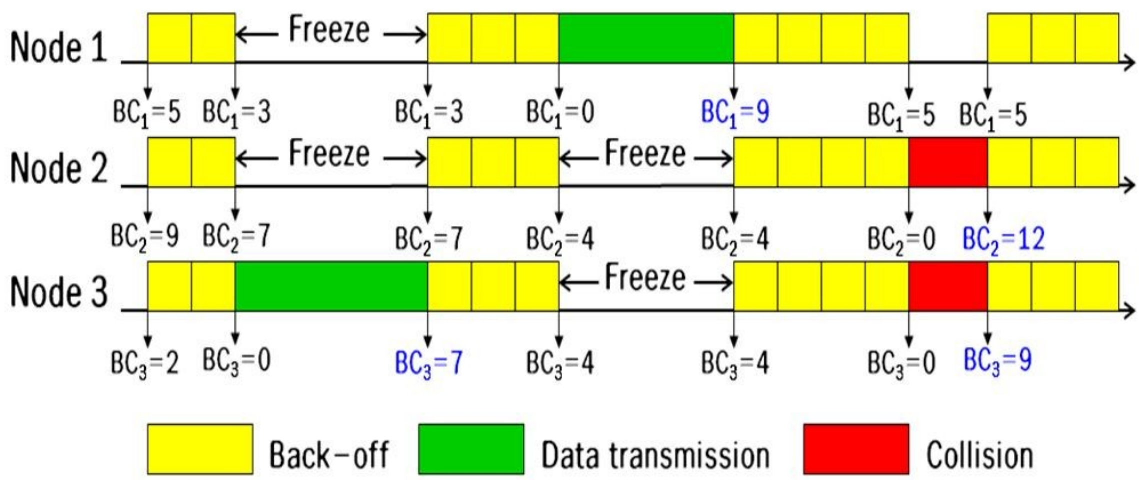


Fig. 3.1. An example of CSMA/CA mechanism

There are basically three parameters of IEEE 802.11 MAC, the initial contention window size (W), the number of backoff stages with doubling (m), and the number of maximum retransmissions (K). K includes the number of backoff stages with no doubling. These MAC parameters affect the wireless network delay distribution.

We consider a two-way handshaking scheme of the DCF as the basic access mechanism for our approach, although there is another mechanism, the four-way handshaking scheme called the RTS/CTS mechanism.

3.3. IEEE 802.11 MAC Access Delay Model

In this subsection, IEEE 802.11 MAC access delay model, which is based on [4], is introduced for the analysis. First, the collision probability p is one of the important elements in the MAC access delay model. In order to get the equation of collision probability p and the attempt probability τ, we first assume p is the same value for every slot and saturated situation. There are n stations to communicate with each other in the model. The collision probability p can be obtained from (9) in [12].

$$p = 1 - (1 - \tau)^{n-1}. \tag{1}$$

The attempt probability τ is a function of p. To obtain τ, we used the two-dimensional Markov chain analysis in [12]. However, a simpler equation is developed using the one-dimensional Markov chain analysis in [13]. The durations of the backoff period are given by discrete uniform random variables.

$$U(i) = \begin{cases} u(0, \lambda^i W - 1) & \text{for } i = 0, \cdots, m-1, \\ u(0, \lambda^m W - 1) & \text{for } i = m, \cdots, K-1. \end{cases} \tag{2}$$

Because u is the uniform distribution, the average backoff duration is

$$\mathrm{E[U(i)]} = \begin{cases} (\lambda^i W - 1)/2 & \text{for i} = 0, \cdots, m-1, \\ (\lambda^m W - 1)/2 & \text{for i} = m, \cdots, K-1. \end{cases} \tag{3}$$

In [13], τ^{-1} can be calculated by the relative frequency π_i and the reciprocal attempt probability as follows.

$$\tau^{-1} = \sum_{i=0}^{K} \pi_i E[U]$$
$$= \frac{(1-p)W(1-(\lambda p)^m)}{2(1-p^K)(1-\lambda p)} + \frac{\lambda^m W(p^m - p^K)}{2(1-p^K)} - \frac{1}{2}. \tag{4}$$

From (1), (4), a fixed point formulation which obtains the answer of *p*, τ can be computed using a numerical technique proposed in [14]. Based on the obtained *p*, τ, we can calculate the MAC access delay. In [4], the random variable *D* is the MAC access delay of a selected station as follows.

$$D = A + T, \tag{5}$$

where the random variable *A* is the sum of collisions and backoff durations and the random variable *T* is the duration of the channel occupancy of a transmitted packet. *A* includes $A(i)$, each transmission probability of when *i* retransmissions (collisions). The probability of *i* retransmissions is ηp^i for $\eta = (1-p)(1-p^K)^{-1}$. The range of *i* is from 1 to *K*-1. $A(i)$ can be written

$$A(i) = \sum_{j=0}^{i} B_i(j) + \sum_{j=1}^{i} C_{ij}. \tag{6}$$

where C_{ij} represents the channel occupancies of collisions, and $B_i(j)$ represents the backoff interval and interruptions of stations. A new representation $B(j)$ for simplicity is developed by dropping index *i*.

$$B(j) = \sum_{k=1}^{U(j)} (t_{slot} + Y_k), \tag{7}$$

where t_{slot} represents a slot time, Y_k is the durations of interruptions. The probability q of at least one transmission by the *n*-1 stations is

$$q = (\mathrm{n}-1)\tau(1-\tau)^{n-2}. \tag{8}$$

There are three cases for representing *Y*. If transmission of unselected stations does not occur in the slot, *Y* is equal to 0. Second, if there is only one transmission with the probability q, *Y* has the random variable T^* which is equal to the channel occupancy of a successful transmission. If there is a collision, *Y* has the random variable C^* represents the channel occupancy of a collision of unselected stations. So, *Y* can be written

$$Y = \begin{cases} 0 & \text{for } 1-p, \\ \mathrm{T}^* & \text{for } q, \\ \mathrm{C}^* & \text{for } p-q. \end{cases} \tag{9}$$

They assume the constant data packet length for the simple equation. Thus,

$$T = t_{data} + t_{difs}, \tag{10}$$

$$C = T^* = C^* = t_{data} + t_{sifs} + t_{ack} + t_{difs}\ for\ C = C_{ij}. \tag{11}$$

where t_{data} denotes the transmission time of a data packet with fixed length, and $\mathrm{t}_{\mathrm{difs},}$ t_{sifs} denote the duration of the DIFS, SIFS. t_{ack} is the duration of the ACK packet transmission. From analysis of the MAC access delay in the previous process, we can find the expression for the average of the MAC access delay E[D] and the variance of the MAC access delay *V[D]* from (18), (19) in [4]. The equations are as follows.

$$E[D] = \eta \sum_{i=0}^{K-1} p^i \{(t_{slot} + E[Y]) \sum_{j=0}^{i} E[U(j) + iE[C]\} + E[T], \tag{12}$$

$$\mathrm{Var[D]} = \eta \sum_{i=0}^{K-1} p^i \left\{ \sum_{\mathrm{j=0}}^{\mathrm{i}} \begin{matrix} (\mathrm{E[U(j)]Var[Y]} + \theta^2 \mathrm{Var[U(j)]}) + \mathrm{iVar[C]} + \\ \left(\theta \sum_{\mathrm{j=0}}^{\mathrm{i}} \mathrm{E[U(j)]} + \mathrm{iE[C]} - \mathrm{E[A]}\right)^2 \end{matrix} \right\} + \mathrm{Var[T]}. \tag{13}$$

where $\theta = t_{slot} + E[Y]$. *Var[T], Var[C], Var[T*], Var[C*]* are equal to zero due to the constant T and C.

To use the MAC delay model in later simulations and experiments, we use the generating function of CCDF of the IEEE 802.11 MAC access delay from (23) in [4]. The simulation parameters of IEEE 802.11b MAC and PHY as listed in Table 3.1. We use these parameters in later simulations and experiments.

Table 3.1. 802.11 b MAC and PHY parameters used in the simulation [4].

Parameter	Symbol	Value	Parameter	Symbol	Value
Data bit rate	r_{data}	11 Mbps	Slot time	t_{slot}	20 us
Control bit rate	r_{ctrl}	1 Mbps	SIFS	t_{sifs}	10 us
PHYS header	t_{phys}	192 us	DIFS	t_{difs}	50 us
MAC header	l_{mac}	224 bits	Min CW	W	32
UDP/IP header	l_{udpip}	320 bits	Doubling limit	m	5
ACK packet	l_{ack}	112 bits	Retry limit	K	7

From the delay model [4], we examine the characteristic of the IEEE 802.11 MAC access delay. Fig 3.2 shows the complementary cumulative distribution function (CCDF) of the MAC access delay as the number of nodes and UDP payloads change. The CCDF is executed for the operation using the LATTICE-POISSON inversion algorithm in [15]. Fig 3.2 shows the CCDF of MAC access delay for n = 10, 30 and the UDP payloads are equal to 33 bytes and 1000 bytes. The CCDF of the MAC access delay depends on the number of stations and UDP payloads. The MAC access delay distribution can have an extremely long access delay value with a very small probability. Consequently, we know the characteristic of the heavy tailed distribution of the IEEE 802.11 MAC access delay.

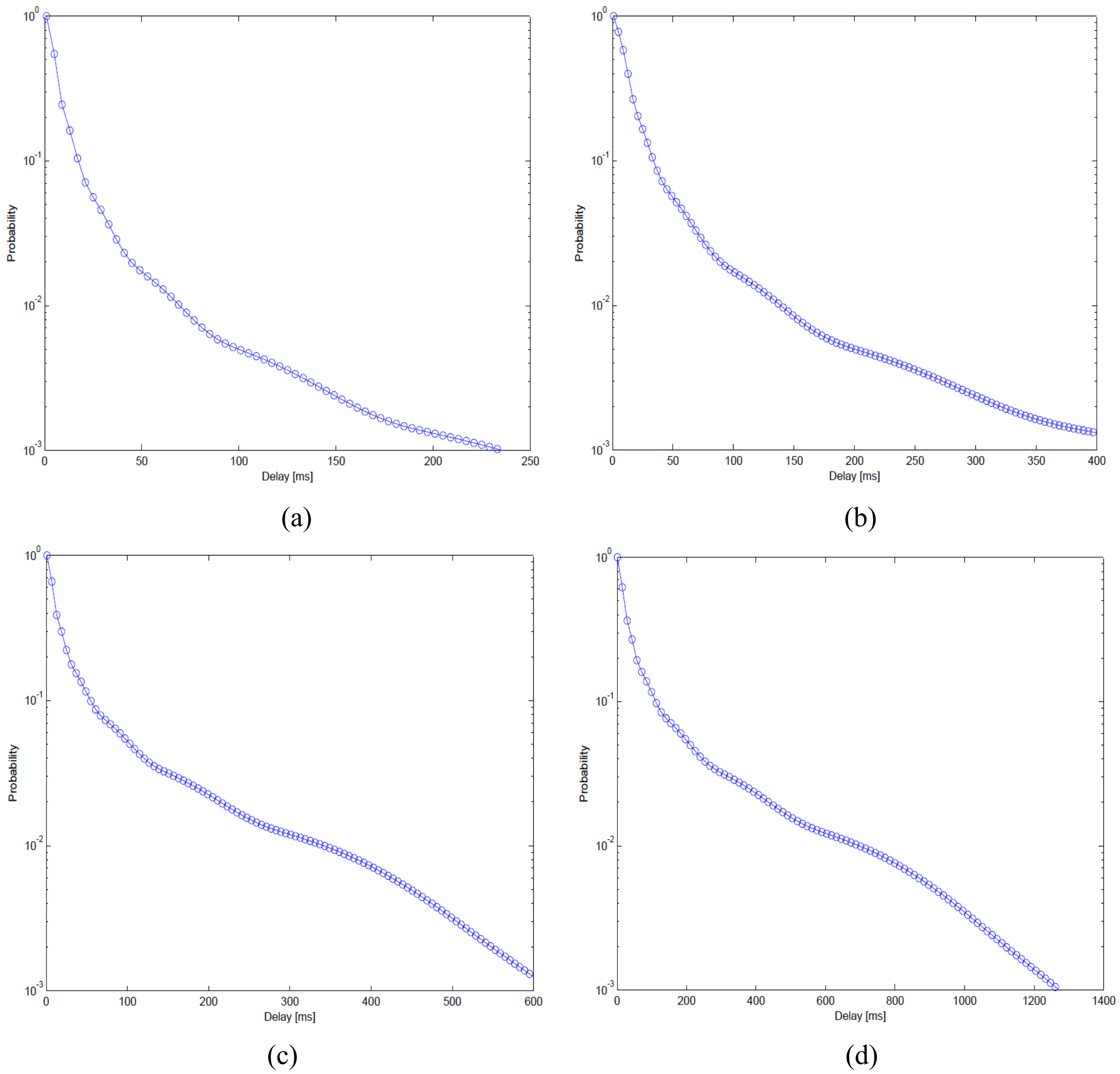


Fig. 3.2. Complementary cumulative distribution function of MAC access delay.

(a) n = 10 and l_{pay} = 33 bytes. (b) n = 10 and l_{pay} = 1000 bytes.

(c) n = 30 and l_{pay} = 33 bytes. (d) n = 30 and l_{pay} = 1000 bytes.

IV. PROBLEM FORMULATION

4.1. Motivation

In the NCS, the network induced delay can have a large effect on NCS performance. There are some solutions in the control side but not much in the network side. Therefore, we focus on solutions of the network side in the NCS as the NCS with wireless network has become popular and various in these days. Therefore, we examine the network delay distribution as IEEE 802.11 MAC parameters which largely affect the delay distribution changes. Through the analysis of the delay model, we determine the effect of each parameter on the delay distribution differently. Finally, we suggest the MAC controller which tunes some MAC parameters in order to optimize the delay distribution for control performance.

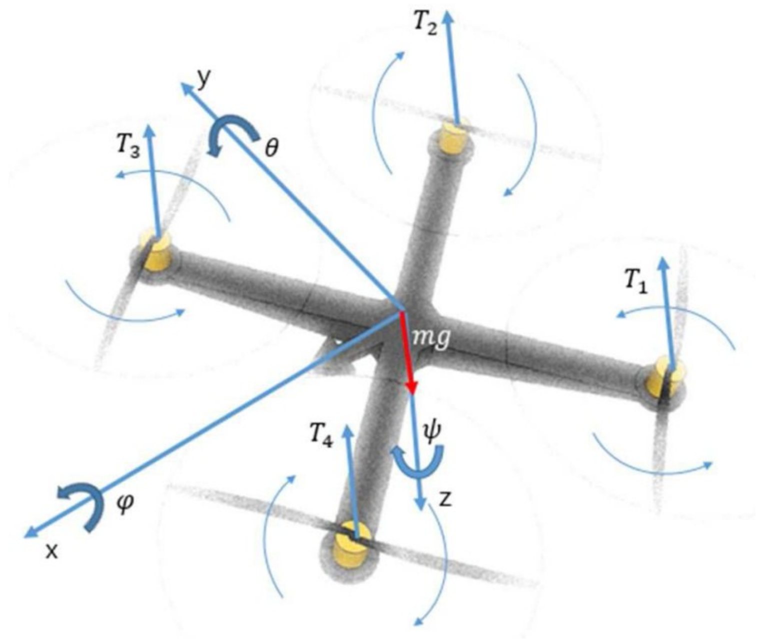


Fig. 4.1. An illustration of a drone.

4.2. System Model

In this subsection, we introduce the system model that verifies our approach through our CPS testbed. First, we introduce the dynamics of a drone as well as the controller design. As shown in Fig. 4.1, the roll angle, the pitch angle, and the yaw angle of the drone are denoted as ϕ, θ, and ψ, respectively. In addition, T_i, i = 1, 2, 3, 4 denote the mechanical force of each motor. The drone has an inner controller, whose inputs are ϕ, θ, ψ and vertical speed [21]. The outputs of the inner controller are the torques of each motor that make the drone posed in the Euler angles of the given data. The overall rigid body dynamics of the drone in Fig. 4.1 is given in [16].

$$
\begin{aligned}
m\ddot{x} &= (\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi)\sum_{i=1}^{4} T_i \,, \\
m\ddot{y} &= (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)\sum_{i=1}^{4} T_i \,, \\
m\ddot{z} &= mg - \cos\theta\cos\phi\sum_{i=1}^{4} T_i \,.
\end{aligned}
\tag{14}
$$

where x, y, and z denote the position of the drone in the corresponding axis. In order to make (14) decoupled, we introduce the following two constraints. First, ψ is fixed at 0 degree, and the height of the drone is fixed to 1 meter. Based on these constraints, we can linearize (14) as follows.

$$
\ddot{x} = \theta g \,, \quad \ddot{y} = -10\phi \,, \quad z = 0 \,. \tag{15}
$$

Since (15) is now decoupled, we can independently control the drone in each axis based on (15). The transfer function from θ to the movement in the x axis and the transfer function from ϕ to the movement in the y axis are given as follows.

$$\frac{X(s)}{\Theta(s)}=\frac{9.8}{s^2},\ \ \frac{Y(s)}{\Phi(s)}=\frac{10}{s^2}. \qquad (16)$$

From (16), we design a proportional-derivative (PD) controller for the x axis and the y axis, and the proportional controller for keeping ψ to zero. The design objective is to make the system have no overshoot and 1.3 sec of 5 % settling time. Then, the gains of the controller are determined as $K_{px}=0.5$, $K_{dx}=0.5$, $K_{py}=0.5$, $K_{dy}=0.5$, $K_{p\psi}=-1$ Here, K_{px}, K_{dx} are gains of the x axis controller, and K_{py}, K_{dy} are gains of the y axis controller, and $K_{p\psi}$ is the gain of the yaw angle controller.

VICON System for Localization

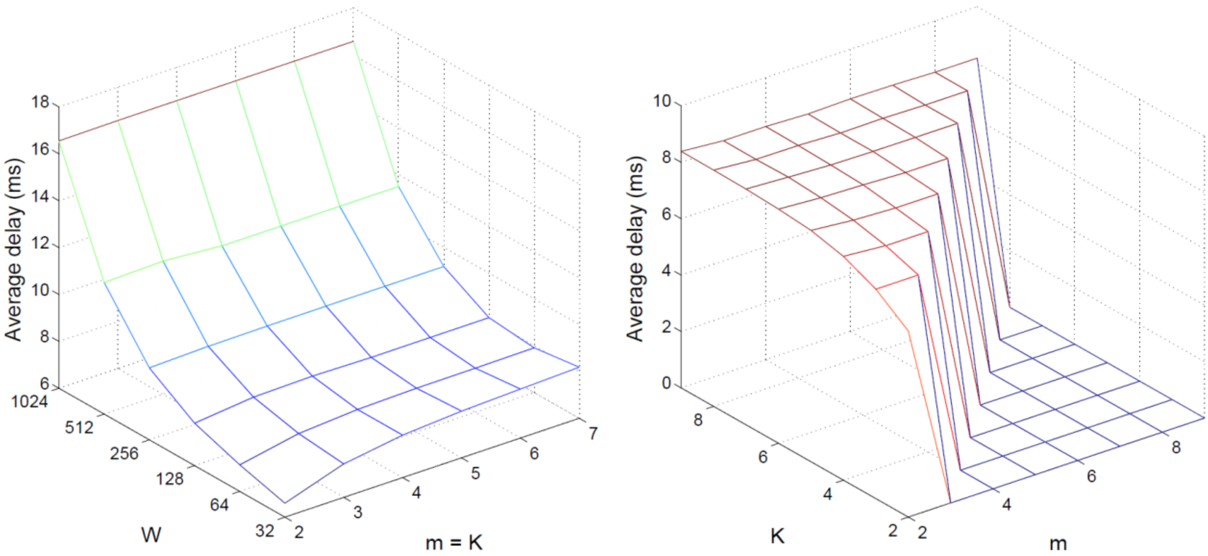
For localization of drones, we use the VICON system [17], which is a state of the art 3D infrared marker tracking system. Our VICON system consists of ten cameras for getting images to construct a three-dimensional representation of markers. Thus, we can know the exact position of the drones. Our overall CPS testbed is already given in Fig. 4.2. The position data of a drone is measured by the VICON system, and the data passes through the SDN to the controller. Then, the controller calculates the control input based on the received data, and sends the control input back to the drone through the SDN. The connection between the controller, VICON system, and the drone is the UDP communication in our testbed. The sampling rate of 200 Hz is used for the VICON system. The VICON system and SDN testbed are shown in Fig. 4.3.

Fig. 4.2. Overall CPS testbed with VICON localization system.


Fig. 4.3. VICON system with a hovering drone.

4.3. Analysis of IEEE 802.11 MAC Delay Model

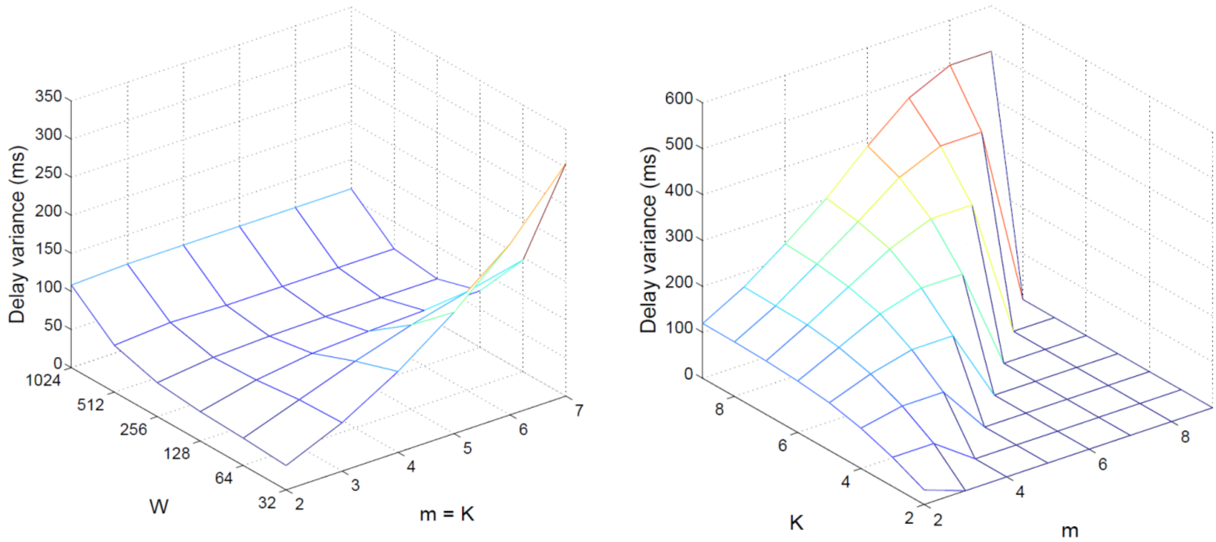
In this section, we show the delay distribution and packet loss according to tuning IEEE 802.11 MAC parameters. After that, we formulate the optimization problem of the delay variance with constrained conditions in the next subsection. We consider a saturated IEEE 802.11 DCF WLAN delay model. With this model, we analyze the impact of the delay distribution and packet loss through 3-dimensional graphs as MAC parameters change. Fig. 4.4(a) and Fig. 4.4(b) display the average delay variation depending on tuning MAC parameters. The initial contention window size (W) has a greater effect on the average delay than other parameters which are almost ineffective.



(a) $m = K$. (b) $W = 32$.

Fig. 4.4. Average delay according to tuning MAC parameters for $n = 10$, l_{pay} = 33 bytes.

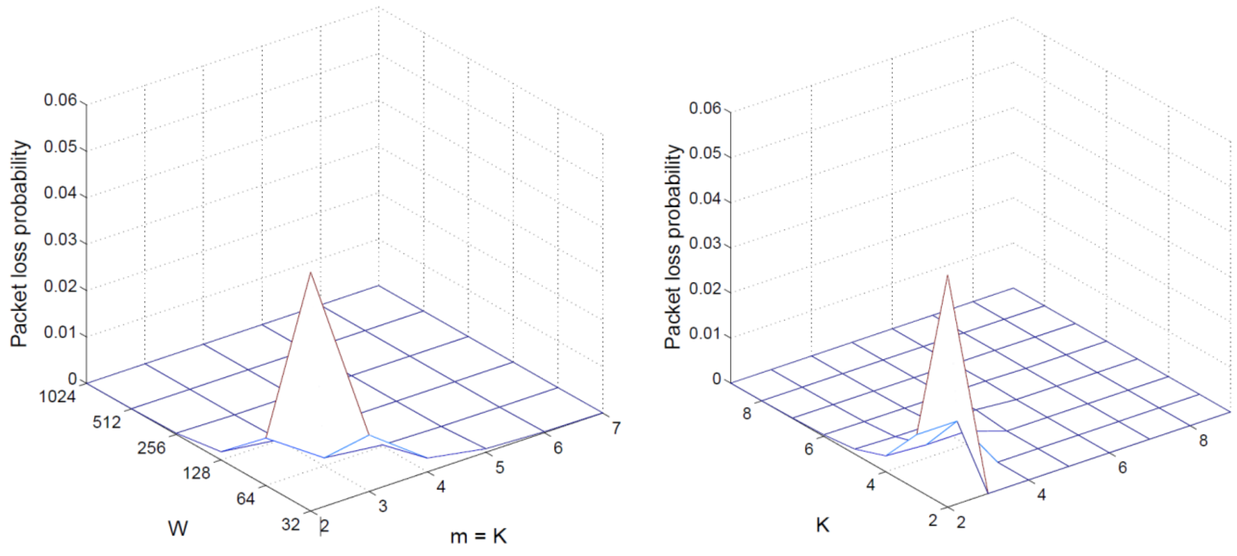
Fig. 4.5(a) and Fig. 4.5(b) show the effect of the delay variance as MAC parameters change. The maximum number of retransmissions K also has an effect on the delay variance as well as W. However, the number of backoff stages (m) has the biggest impact on the delay variance.



(a) $m = K$.　　　(b) $W = 32$.

Fig. 4.5. Delay variance according to tuning MAC parameters for $n = 10$, l_{pay} = 33 bytes.

Fig. 4.6(a) and Fig. 4.6(b) show the packet loss probability. Relatively small values of W, m, K increases the packet loss probability exponentially. Comprehensively, we first have to find a proper W for the average delay, and then the optimized m is needed for the delay variance considering the packet loss probability. The average delay is one of the most important factors for the control performance. However, we consider the delay distribution by tuning IEEE 802.11 MAC parameters, which means we focus on the delay variance as well as the average delay.



(a) $m = K$. (b) $W = 32$.

Fig. 4.6. Packet loss probability according to tuning MAC parameters for $n = 10$, l_{pay} = 33 bytes.

4.4. Optimization Problem Formulation

In this section, we formulate an optimization problem of minimizing the average absolute tracking error with constraints on maximum packet loss, average delay, and delay variance bound. Here, we consider a saturated IEEE 802.11 WLAN delay model, similar to that in [4]. Our objective is to minimize the average absolute tracking error by reducing the average delay and the variance of MAC access delay while maintaining the packet loss below thresholds. In this manner, we can provide better control performance while providing a certain level of system stability. The overall formulation is as follows:

$$\underset{M}{minimize}\ TE(M) \tag{17}$$

$$subject\ to\ L(M) \le L_{max}, A(M) \le A_{max}, V(M) \le V_{max}, M \in \mathcal{M},$$

where $TE(\cdot)$, $V(\cdot)$, $L(\cdot)$, $A(\cdot)$ denote the average absolute tracking error, the delay variance, packet loss, and average delay, respectively. In addition, the decision variables $M = (W, m, K)$ are component-wisely the initial contention window size, the number of backoff stages, and the maximum number of retransmission, respectively, and $\mathcal{M}$ is the feasible set of M.

V. MAC CONTROLLER DESIGN

We propose a MAC controller to adjust MAC parameters every particular period in the NCS. Our design goal for the MAC controller is that NCS improves the control performance by reducing the average absolute tracking error. Thus, the initial contention window size (W) which mainly reduces the average delay is changed and the number of backoff stages (m) which is related to the delay variance is subsequently changed. In our approach, to reduce the complexity of the algorithm, we assume that one or more nodes which have the same priority communicate with the controller through control packets, and other nodes just work normal communication. Also, a MAC controller exists in the device of the controller and it can control MAC parameters over the wireless network from the controller through MAC control packets. Fig. 5.1 shows the NCS architecture with a MAC controller.

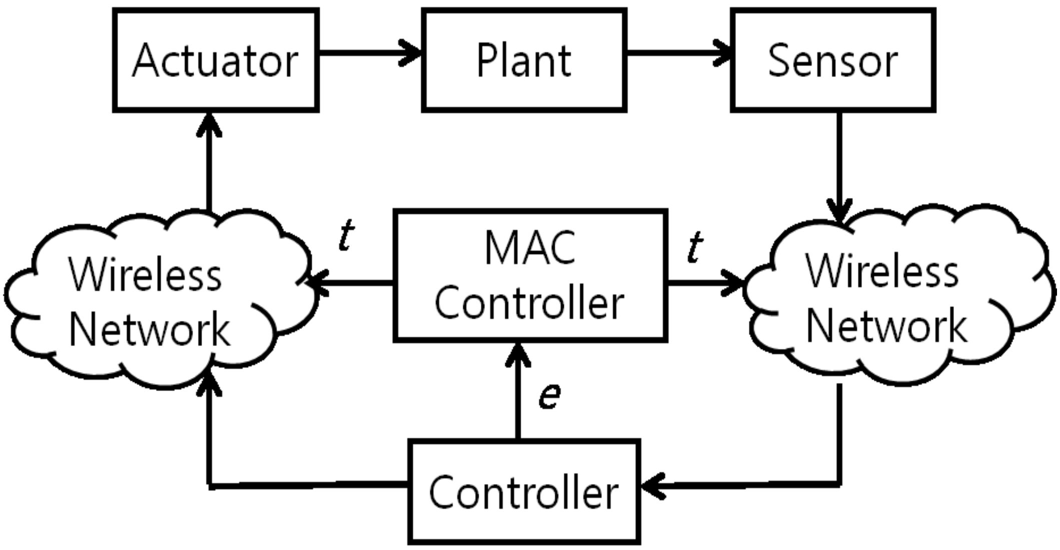


Fig. 3.1. Networked control systems (NCS) architecture with MAC controller.

The MAC controller receives the tracking error data from the controller in the NCS and collects the latest sample data in the memory, and computes the absolute average tracking error in real time. The MAC controller collects the more sample data, finding the optimal MAC parameters is the more accurate. On the other hand, it takes longer to find the MAC optimal parameters. So, there is a tradeoff between accuracy and quickness. One important thing in the MAC controller is how the MAC controller works. We consider the MAC controller operates in three ways.

i) Setting the threshold of the absolute average tracking error.

ii) Operating every regular period.

iii) Comparing prior absolute average tracking error values.

i) When the MAC controller monitors whether the current absolute average tracking error is higher than the predefined threshold, it works to change the optimal MAC parameter according to our algorithm.

ii) In order to keep the optimal control performance in real time, the MAC controller needs to operate periodically even though the absolute average tracking error is lower than threshold.

iii) If the overall network situations are too bad or good, the threshold can be ignored. Thus, comparing the last absolute average tracking error over a longer time, the MAC controller can decide to work continuously.

Our algorithm, which finds the optimal MAC parameters, is simple and based on the analysis of MAC delay model. we first tune W for the decrease of the average delay. After that, the MAC controller changes m to reduce the delay variance and find proper values.

VI. PERFORMANCE EVALUATION

6.1. Preliminary Simulation and Experiment

Prior to applying our approach, we first verify the preliminary simulations and experiments for the effect of the delay variance in the NCS so as to be sure of the conditions of optimal parameters. The next subsection shows the reduction of the delay variance affects the control performance by tuning the number of backoff stages (m). After that, we finally evaluate the control performance through simulations due to the practical limitation.

6.1.1. Preliminary Simulation

We carry out a simulation study of the network control of a drone with the IEEE 802.11 MAC delay model using MATLAB/Simulnk. The overall system is shown in the previous section Fig. 4.2. A PD controller is used for the simulation. The sampling period h is set to 0.01 s in all simulations. In this simulation, the output of the plant tracks the reference input of r = 1. The packet size is l = 33 bytes, and the number of nodes is n = 10. If the control packet does not arrive within the sampling period, prior packet data is used once again. When a later control packet arrives earlier than a prior packet, the prior packet data becomes invalid.

To know the effect of the delay variance, we set the parameter W = 64 for the average delay optimization in this case. At a basic MAC parameter K = 7, we generate the delay distributions on the basis of the IEEE 802.11 MAC delay model. However, we do not consider a limited CCDF of access delay, up to 10^{-10} statistically because CCDF of access delay is a heavy tailed access delay. The average delays with m = 4, 5, 6, which are satisfying constraints, are all 8.4 ms, and the delay variances are 125.4, 137.4, and 144.0 as shown in Table 2. We will compare the control performance of the system in terms of the absolute tracking error according to the step response depending on a parameter m, which means the delay variances are changed.

Table 6.1. Variance table for $m = 4, 5, 6$

	Variance
$m = 4$	125.4
$m = 5$	137.4
$m = 6$	144.0

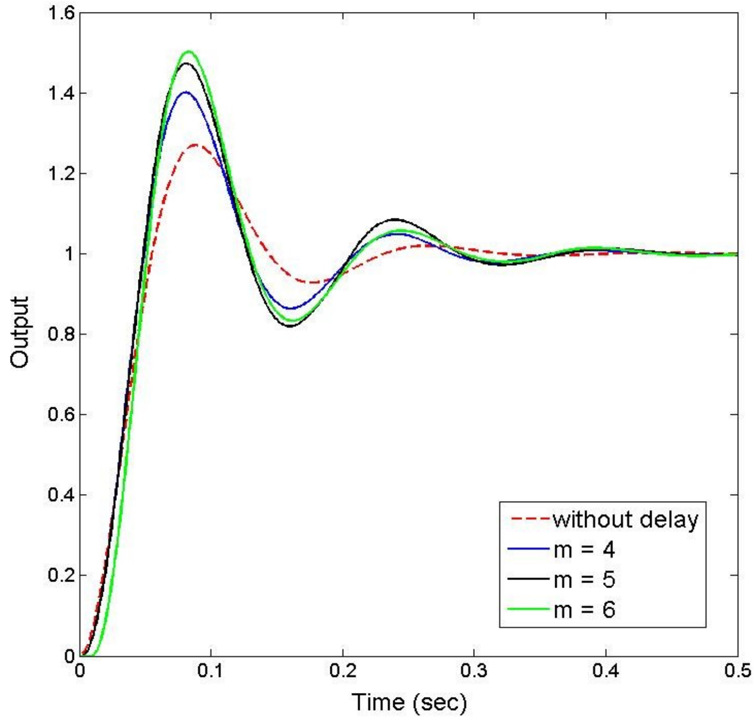


Fig. 6.1. Effect of the delay variance with different MAC parameters of $m = 4, 5, 6$.

As shown in Fig. 6.1, the simulation results show that the controller gives a larger overshoot and settling time as the MAC parameter m increases. Our preliminary result gives a parameter set of $W = 64$, $m = 4$, $K = 7$ for $n = 10$, $l = 33$ bytes. Fig. 6.2 shows specific values of the average absolute tracking error up to 0.5 s, which are 0.1107, 0.1279, 0.1451, and 0.1515, respectively. When there are no network-induced delays in the control system, the tracking error is the smallest. Our simulation results show the absolute tracking error increases as the parameter *m* increases, which means that the variance of the network induced delay increases.

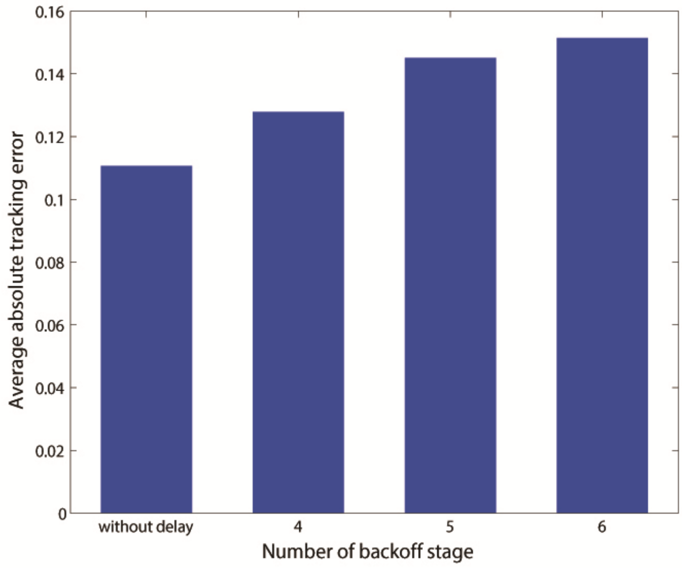


Fig. 6.2. Average absolute tracking error values of the controller.

6.1.2 Preliminary Experiment

We also verify the simulations through the preliminary experiments in our CPS testbed. With a drone and VICON system, the network delay model based function generates delays additionally in the controller. The experimental environment is a little different from simulations because simulation results can have more effect on external factors such as relatively short delays. New MAC parameters are set for W = 128, n = 20 and other conditions are the same. There are two kinds of reference input, the step input and the sine wave input. Fig. 6.3 shows the change of the real *x*/*y* positions about the reference signal without the network-induced delay. The *x* axis and *y* axis are the *x* and *y* positions, respectively. The unit of each axis is the meter and a drone moves from (1 , 1) to (-1, -1) during 8 *s*. The blue line is the reference input, and the red line is real position of a drone. Figs. 6.4(a), (b) represent the change of the *x*/*y* positions, which are separated by the time and use the same controller. The *x* axis is the time up to 8 *s*, and the *y* axis is the position. There are the transient error and steady state error of the output. The settling time is approximately 3.3 *s* by a transient error. After that, the steady state error is almost nonexistent. Fig. 6.4(c) shows the change of the tracking error as time passes. Comparing a drone system without a network-induced delay to a drone system with a network-induced delay, we examine how the error performance differs.

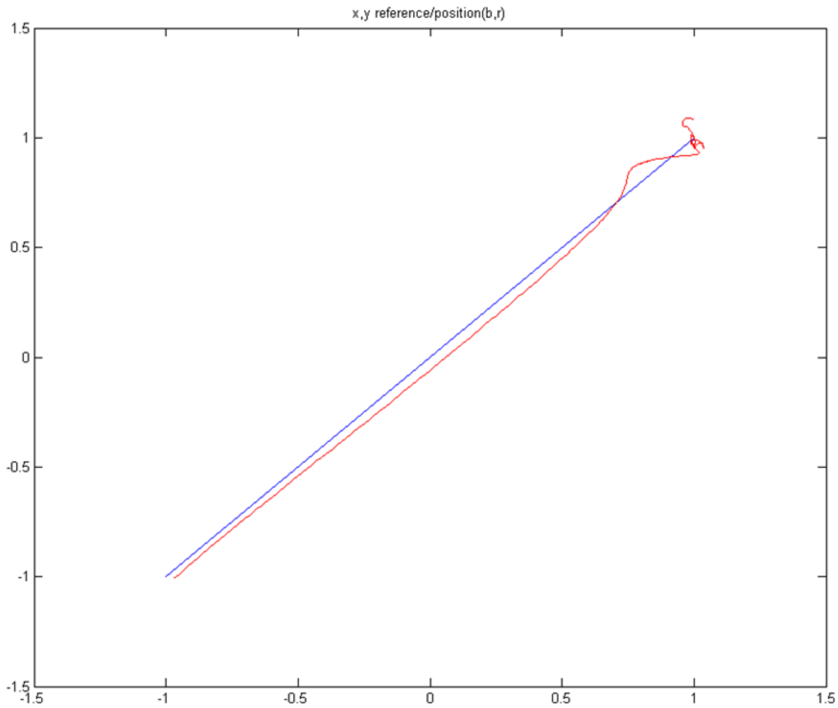
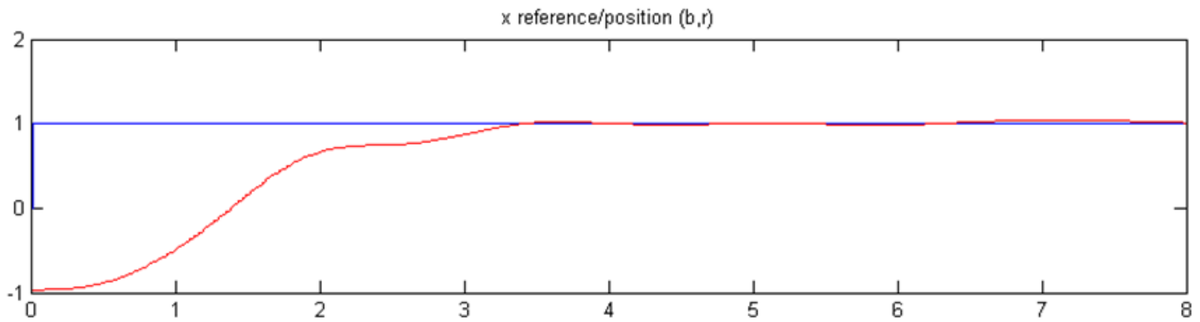
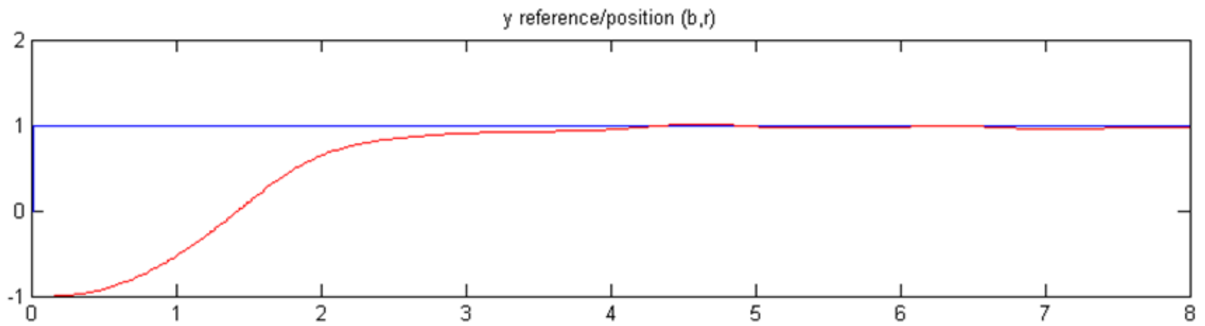


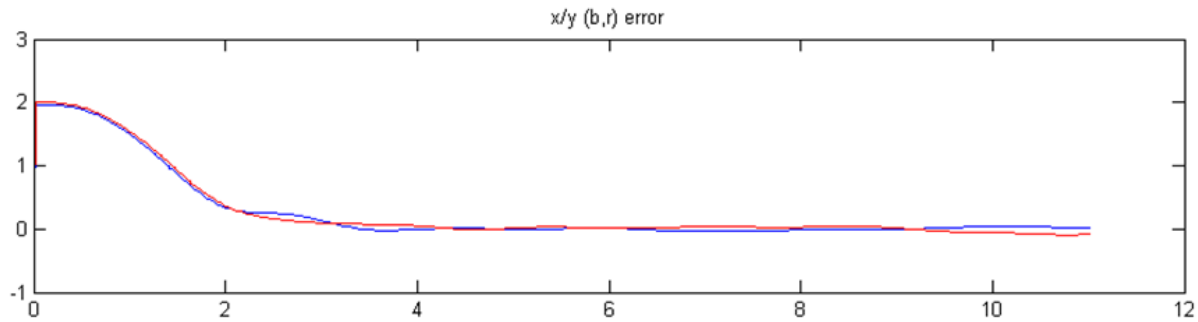
Fig. 6.3. *x*/*y* reference signal and position change (blue, red) without the network delay.



(a) *x* reference signal and position (blue, red).



(b) *y* reference signal and position (blue, red).



(c) The change of *x*/*y* (blue, red) tracking error.

Fig. 6.4. Step input, responses and absolute error of a drone without the network delay.

The change of reference input and real position with the network induced delay in Fig. 6.5 when the number of backoff stages (m) is equal to 4. As expected, the overall tracking error increases remarkably compared to no delay. As shown in Fig. 6.6(a), (b), the settling time is 4.4 s and takes longer by about 1 s in comparison with a drone without a network delay. In addition, the steady state wave is more rolling in this case. Fig. 6.6(c) shows the tracking error for the step response, which is also larger than the drone system with the delay. Thus, we verify experimentally that the effect on the network induced delay is quite large in the control system.

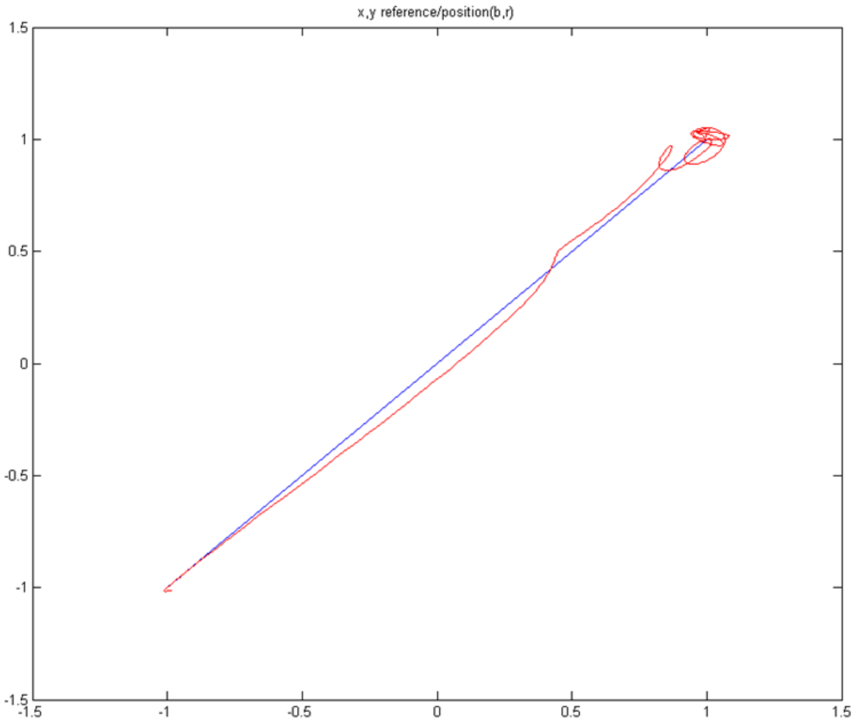
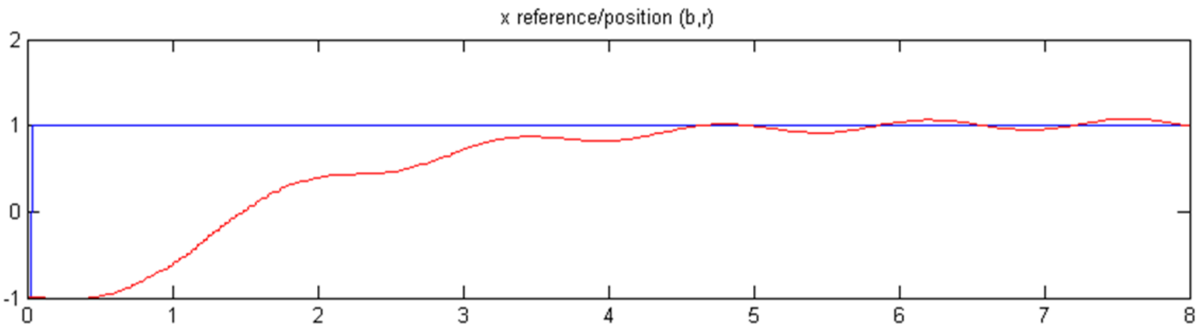
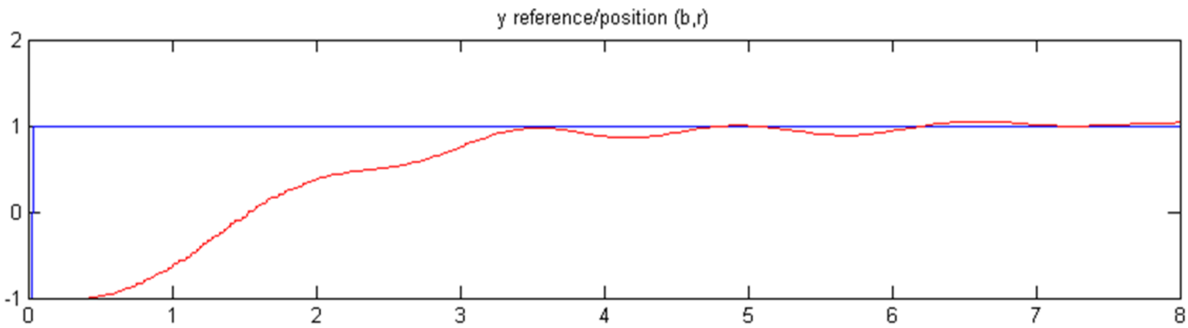


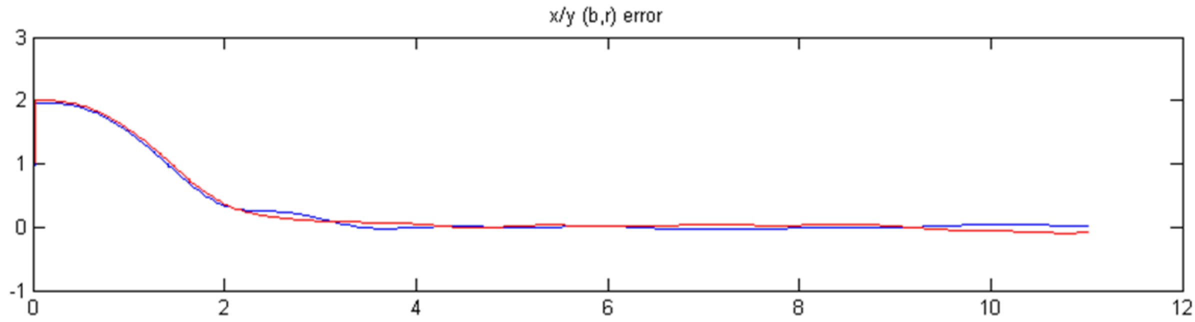
Fig. 6.5. *x*/*y* reference signal and position change (blue, red) with the network delay for $m = 4$.



(a) *x* reference signal and position (blue, red).



(b) *y* reference signal and position (blue, red).



(c) The change of *x*/*y* (blue, red) tracking error.

Fig. 6.6 Step input, responses and absolute error of a with the network delay for $m = 4$.

Table 6.2 represents the experimental summary of the absolute average tracking error for m = 4, 5, 6 as well as no delay case when the reference signal is a step input, which is used up to 8 *s*. Experimental results show that a lower *m* makes a lower absolute average tracking error because the delay variance is lower.

Table 6.2. Absolute average tracking error for step response

	Absolute average tracking error
No delay	0.7794
$m = 4$	0.9361
$m = 5$	0.9559
$m = 6$	0.9918

Fig. 6.7 represents the *x*/*y* positions of a drone with the network delay for $m = 4$ up to 60 *s* when the reference input is $1.5\sin(\frac{w}{16}t)$ as sine wave. As shown in Figs. 6.7(a), (b), the overall tracking error is lower than the previous step response as the reference signal changes slowly. The exact figure is shown in Table 6.3. However, as the parameter *m* changes, the tendency is similar to the step input case. Consequently, we need to consider the effect of the delay variance.

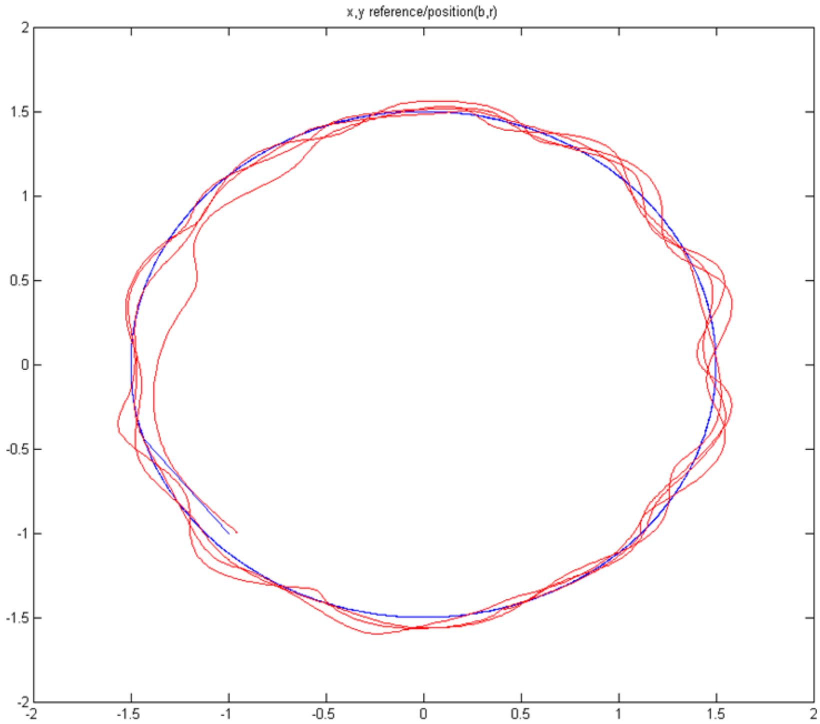
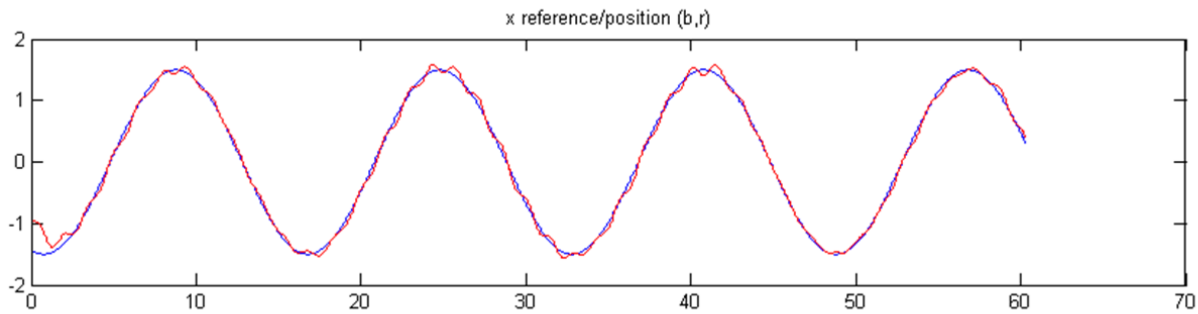
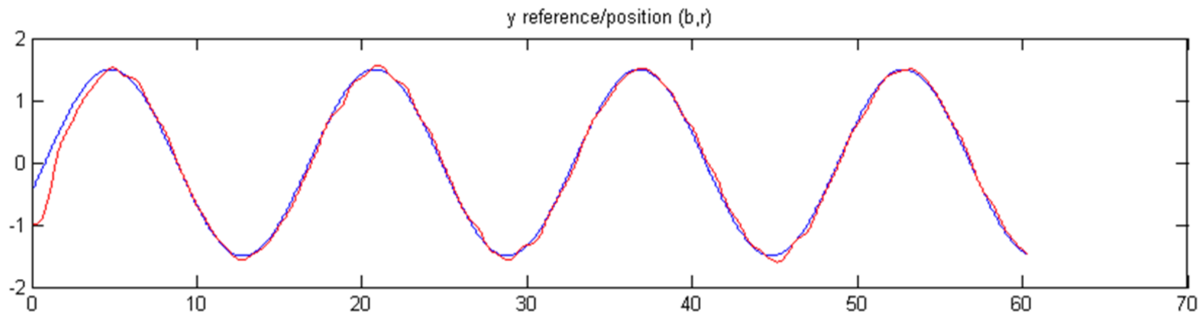


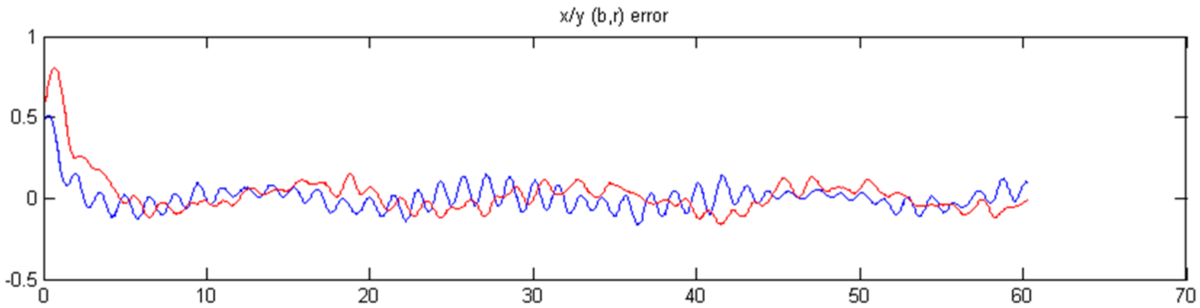
Fig. 6.7. *x*/*y* reference signal and position change (blue, red) with the network delay for $m = 4$.



(a) *x* reference signal and position (blue, red).



(b) *y* reference signal and position (blue, red).



(c) The change of *x*/*y* (blue, red) tracking error.

Fig. 6.8. Sine wave input, responses and absolute tracking error with the network delay for m = 4.

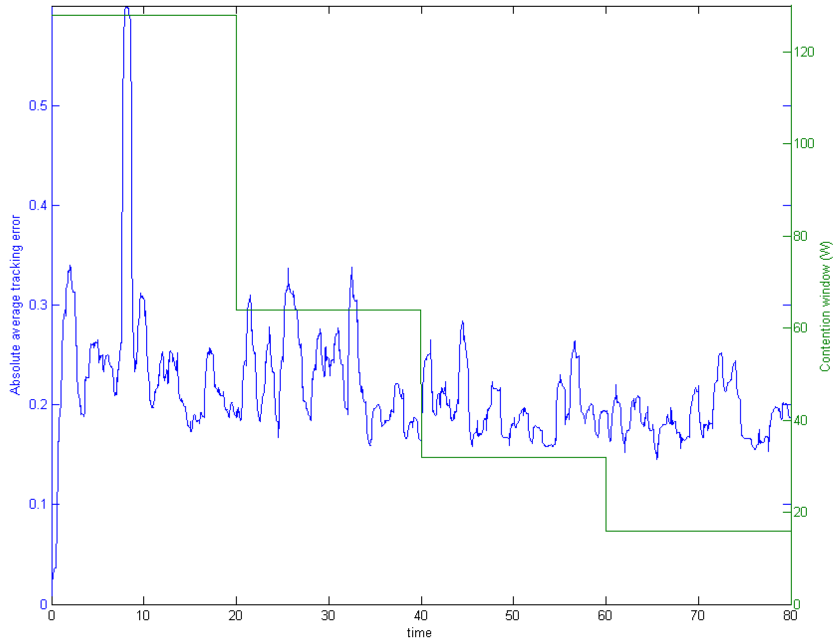
Table 6.3. Absolute average tracking error for sine wave

	Absolute average tracking error
m = 4	0.1032
m = 5	0.1297
m = 6	0.1542

6.2. MAC Control Performance Evaluation

In this subsection, we verify the performance of the MAC controller. We carry out a the MAC control simulation using MATLAB/Simulnk. Initial simulation parameters are set for W = 128, m = 6, K = 7, n = 10, l_{pay}= 33 bytes. The reference input is the square wave with an amplitude of 1 and frequency of 1Hz. When the MAC controller is operating, the MAC parameter is changed every 20 s. Our algorithm of the MAC controller is as follows.

First, we just tune W in order to optimize the average delay. Fig. 6.9 represents the simulation result when the parameter W was changed from 128 to 16 by the MAC controller. During the first 2 s, the MAC controller stores the initial 200 samples from the controller. After that, normal average calculation is processed. As shown in Table 6.4, the absolute average tracking error improves as W decreases. As W becomes smaller, the packet loss probability has to become higher. However, due to the small number of nodes and data payloads, the effect of the packet loss becomes small. Therefore, we know the optimal parameter W is 16 in this case.



.

Fig. 6.9. Absolute average tracking error and W when the MAC controller operates.

Table 6.4. Absolute average tracking error of Fig. 6.9 with the MAC controller every 20 *s*

	Absolute average tracking error
0-20 s (W = 128)	0.2475
20-40 s (W = 64)	0.2278
40-60 s (W = 32)	0.1972
60-80 s (W = 16)	0.1875

Second, we tune m to optimize the delay variance. Fig. 6.10 shows a simulation result as the MAC parameter m changed every 20 *s* from 6 to 3 by the MAC controller. As shown in Table 6.5, the values of absolute average tracking error look like a convex relationship. In [5], the convex relationship between the retransmission limits was revealed through an experimental study. We find the optimal parameter m = 5.

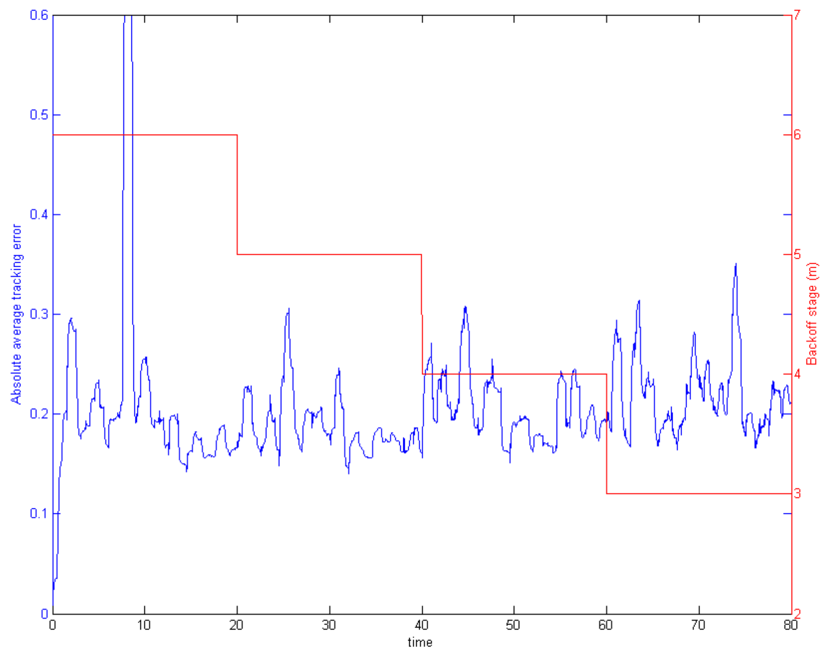


Fig. 6.10. Absolute average tracking error and m when the MAC controller operates.

Table 6.5. Absolute average tracking error of Fig. 6.10 with the MAC controller every 20 *s*

	Absolute average tracking error
1-20 s (m = 6)	0.2133
20-40 s (m = 5)	0.1881
40-60 s (m = 4)	0.2064
60-80 s (m = 3)	0.2199

From the simulation results, we find the optimal MAC parameter of W, m in this case. In various situations, the MAC controller determines the optimal MAC parameter. Consequently, we can guarantee the control performance from the network error in the CPS.

VII. CONCLUSION

We have evaluated the control performance with respect to the variance of the IEEE 802.11 MAC access delay by tuning key MAC parameters. Our main idea was to take into account the delay distribution rather than simply considering the average delay for better control performance in a CPS. We found that the MAC parameter has a large effect on the average delay and the delay variance in order to improve the average absolute tracking error as a performance index. We also presented a MAC controller that controls the MAC parameters by our algorithm. Our simulation results show that our approach can give better control performance.

REFERENCE

[1] R. Poovendran, K. Sampigethaya, S. K. S. Gupta, I. Lee, K. V. Prasad, D. Corman, and J. L. Paunicka, “Special issue on cyber-physical systems [scanning the issue],” *Proceedings of the IEEE*, vol. 100, no. 1, pp. 6–12, Jan. 2012.

[2] K.-J. Park, R. Zheng, and X. Liu, “Cyber-physical systems: Milestones and research challenges,” *Computer Communications*, vol. 36, no. 1, pp. 1–7, Dec. 2012.

[3] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, “A survey of recent results in networked control systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, Jan. 2007.

[4] T. Sakurai and H. L. Vu, “MAC access delay of IEEE 802.11 DCF,” *IEEE Transactions on Wireless Communications*, vol. 6, no. 5, pp. 1702–1710, 2007.

[5] J. Bai, E. P. Eyisi, Y. Xue, and X. D. Koutsoukos, “Dynamic tuning retransmission limit of ieee 802.11 mac protocol for networked control systems,” in *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int’l Conference on & Int’l Conference on Cyber, Physical and Social Computing (CPSCom). IEEE*, 2010, pp. 666–671.

[6] M. S. Branicky, S. M. Phillips, and W. Zhang, “Stability of networked control systems: Explicit analysis of delay,” in American Control Conference, 2000. Proceedings of the 2000, vol. 4. IEEE, 2000, pp. 2352–2357.

[7] B. Ramakrishnan, R. Rajesh, and R. Shaji, “Performance analysis of 802.11 and 802.11 p in cluster based simple highway model,” *International Journal of Computer Science and Information Technologies*, vol. 1, no. 5, pp. 420–426, 2010.

[8] P. Park, P. Di Marco, C. Fischione, and K. H. Johansson, “Delay distribution analysis of wireless personal area networks,” in Proc. IEEE CDC, 2012, pp. 5864–5869.

[9] P. Park, P. Di Marco, C. Fischione, K. Johansson, “Modeling and Optimization of the IEEE 802.15.4 Protocol for Reliable and Timely Communications”, IEEE Transactions on Parallel and Distributed Systems, 2012.

[10] T Madhavi , G Sasi Bhushana Rao , M Rajan Babu , K Sridevi, Analysis of Throughput and Energy Efficiency in the IEEE 802.11 Wireless Local Area Networks using Constant backoff Window Algorithm, International Journal of Computer Applications, 2011,Vol: 26 Issue: 8 ,40-47.

[11] W. Wang, S. Liew, and V. O. K. Li, “Solutions to performance problems in VoIP over a 802.11 wireless LAN,” IEEE Trans. Veh. Technol., vol. 54, no. 1, pp. 366–384, Jan. 2005.

[12] G. Bianchi, “Performance analysis of the IEEE 802.11 distributed coordination function,” IEEE J. Sel. Areas Commun., vol. 18, pp. 535– 547, 2000.

[13] B.-J. Kwak, N.-O. Song, and L. E. Miller, “Performance analysis of exponential backoff,” IEEE/ACM Trans. Networking, vol. 13, pp. 343– 355, 2005.

[14] A. Kumar, E. Altman, D. Miorandi, and M. Goyal, “New insights from a fixed point analysis of single cell IEEE 802.11 WLANs,” in Proc. IEEE INFOCOM 2005, pp. 1550–1561.

[15] J. Abate, G. L. Choudhury, and W. Whitt, “An introduction to numerical transform inversion and its application to probability models,” in Computational Probability, W. K. Grassman, Ed. Norwell, MA: Kluwer, 2000, pp. 257–323.

[16] J. Pestana, J. L. Sanchez-Lopez, I. Mellado-Bataller, C. Fu, P. Campoy. "AR Drone identification and navigation control at CVG-UPM," In XXXIII Jornadas de Automática, September 2012.

[17] VICON System. http://www.vicon.com.

[18] M. S. Branicky, S. M. Phillips, and W. Zhang, “Stability of networked control systems: Explicit analysis of delay,” in American Control Conference, 2000. Proceedings of the 2000, vol. 4. IEEE, 2000, pp. 2352–2357.

[19] IEEE Computer Society LAN MAN Standards Committee. IEEE Standard for Wireless LAN Medium Access Control and Physical Layer specifications, 1997. Project IEEE 802.11.

[20] Q. Zhao, D.H.K. Tsang, and T. Sakurai, “A Simple and Approximate Model for Nonsaturated IEEE 802.11 DCF,” IEEE Trans. Mobile Computing, vol. 8, no. 11, pp. 1539-1553, Nov. 2009.

[21] S. Piskorski, N. Brulez, P. Eline, and F. DHaeyer, AR.Drone Developer Guide, Parrot, December 2012, SDK Version 2.0.

[22] C. Lu, R. Rajkumar, and E. Tovar, “Guest editorial: Special section on cyber-physical systems and cooperating objects,” IEEE Trans. Ind. Informat., vol. 8, no. 2, p. 378, May 2012.

[23] T. C. Yang, “Networked control system: A brief survey,” Proc. Inst. Elect. Eng., vol. 153, no. 4, pp. 403–412, Jul. 2006.

요 약 문

사이버-물리 시스템의 제어 성능 향상을 위한 IEEE 802.11 MAC 액세스 접근 딜레이 조정

본 논문에서는 사이버-물리 시스템의 제어 성능을 향상시키기 위해 무선 네트워크인 IEEE 802.11 을 사용하는 제어 시스템에서 MAC (Medium Access Control) 계층의 딜레이 파라미터를 조정함으로써 최적화된 제어 성능을 유지하고자 한다. 이 논문의 주된 아이디어는 IEEE 802.11 MAC 딜레이 모델 기반으로 MAC 파라미터 변화에 따른 네트워크 딜레이 분포를 고려하여 실시간으로 제어 성능을 최대한 보장하는 알고리즘을 제안하는 것이다. 기존 네트워크가 포함된 제어 시스템에서는 단순히 가우시안 랜덤 딜레이를 고려하여 시스템을 설계하는 데, 실제 무선 네트워크의 딜레이 분포는 꼬리가 긴(long-tailed) 형태의 특징을 가지므로 모델의 정확성이 떨어진다. 따라서, 가장 널리 쓰이는 무선 네트워크 프로토콜 중 하나인 IEEE 802.11 의 MAC 딜레이 모델을 분석하고 딜레이의 평균과 분산을 모두 고려하여 최적화된 MAC 파라미터를 찾는 방법을 제안한다.

이에 앞서, 사이버 물리 시스템의 제어 성능을 보장하기 위한 최적화 문제를 소개한다. 또한, MAC 딜레이 분산에 따른 제어 성능의 영향을 소개하고 유효성을 검증하기 위해 무선 네트워크가 포함된 드론과 VICON 시스템으로 사이버-물리 시스템 테스트베드 모델을 바탕으로 시뮬레이션, 실제 구축 및 실험을 통해 확인한다. 제어 성능의 비교는 레퍼런스 신호에 따른 컨트롤러의 절대 평균 에러를 측정하여 성능을 비교한다.

마지막으로 MAC 파라미터를 조정하기 위해 실시간으로 딜레이 분산의 영향을 최소화시키는 MAC 컨트롤러를 제안한다. MAC 컨트롤러는 제어기의 절대 평균 에러와 딜레이의 분포를 통해 동작하며 제안된 알고리즘에 의해 파라미터를 조정한다. 시뮬레이션을 통해 검증이 되며 결과는 제안한 방법으로 제어 성능을 향상시킬 수 있음을 보여준다.

핵심어: Cyber-physical systems, MAC access delay, IEEE 802.11, control performance