JPNT 10(4), 387-393 (2021)
https://doi.org/10.11003/JPNT.2021.10.4.387

JPNT Journal of Positioning, Navigation, and Timing

Synchronization System for Time of Mission and Flight Computers over UAV Network

Won-Seok Lee[1,2], Jun-Yong Jang[1,2], Hyoung-Kyu Song[1,2†]

[1]Department of Information and Communication Engineering, Sejong University, Seoul 05006, Korea
[2]Department of Information and Communication Engineering, Convergence Engineering for Intelligent Drone

ABSTRACT

This paper proposes a system to synchronize the time of computers over an unmanned aerial vehicle (UAV) network. With the proposed system, the UAVs can perform missions that require precise relative time. Also, data collected by UAVs can be fused precisely with synchronized time. In the system, to synchronize the time of all computers over the UAV network, two-step synchronization is performed. In the first step, the mission computers of the UAVs are synchronized through the server of the system. After the first step, the mission computers measure time offset between the time of the mission computers and the flight computers. The offset values are delivered to the server. In the second step, virtual time is determined by the server from the collected time offset. The measured offset is compensated by moving the synchronized time of mission computers to the reasonable virtual time. Since only the time of mission computers are controlled, any flight computers that use micro air vehicle link (MAVLink) protocol can be synchronized in the proposed system.

Keywords: UAV, time synchronization, NTP, Pixhawk, Raspberry Pi

1. 서론

시간 동기화는 네트워크 분야에서 중요한 이슈 중 하나로 네트워크를 구성하는 노드와 네트워크의 구조 및 목적에 따라 다양한 기술이 제안되었다 (Lasassmeh & Conrad 2010). 1980년대에 등장한 후 현재까지 인터넷에서 사용되는 가장 유명한 동기화 기술은 network time protocol (NTP) 이다 (Mills 1991, Mills et al. 2010, Lasassmeh & Conrad 2010, Li et al. 2019). NTP는 인터넷에 접속한 컴퓨터들의 시간을 수 밀리 초 수준의 오차내로 동기화하기 위해 만들어졌다. NTP의 목적은 다양한 연결 경로 및 네트워크 구조가 존재하는 인터넷에서 시간에 대한 신뢰를 보장할 수 있는 데이터 교환을 지원하는 것이다. 일반적으로 인터넷을 통한 데이터 교환에서 시간에 대한 신뢰를 보장하는데 요구되는 정밀도는 NTP가 제공하는 성능으로도 충분하기 때문에 현재까지도 문제없이 사용되고 있다. 마이크로 혹은 나노 초 단위의 시간 동안 네트워크를 이용한 실험을 수행하는 연구는 높은 정밀도의 시간 동기화 기술이 필요하다. 이러한 네트워크는 일반적으로 유선으로 구현되며 동기화 알고리즘을 보완하기 위한 다양한 부가적 기술들이 사용된다. Precision Time Protocol (PTP)는 높은 정밀도의 시간동기화를 보장하는 대표적인 프로토콜이며, 동기화 알고리즘은 NTP와 동일하지만 하드웨어 단계의 타임스탬프를 교환하는 방법으로 구현하여 마이크로 초 수준의 시간 동기화가 가능하다 (Li et al. 2019).

무선 네트워크는 구성 노드들이 통신 채널을 확보하고 패킷을 전달하는 시간이 유선 네트워크보다 불안정하다 (Elson et al. 2002, van Greunen & Rabaey 2003, Johannessen 2004, Solis et al. 2006, Hasan et al. 2012). 그렇기 때문에 유선 네트워크가 달성가능한 시간 동기화의 정밀도보다 상대적으로 낮은 수준의 정밀도를 달성할 수밖에 없다. 무선 네트워크의 또 다른 문제는 네트워크를 구성하는 노드들이 베터리로 동작하는 경우가 대부분이라는 것이다. 그래서 무선 네트워크를 시간 동기화 프로토콜은 전력을 적게 소비하는 것을 중요한 문제로 다룬다. 무선 네트워크는 단일 연결에서 발생하는 전송 과정의 변동성 및 전력의 제한과 같은 문제가 있지만 네트워크의 규모가 유선 네트워크에 비

Received Oct 18, 2021 **Revised** Oct 27, 2021 **Accepted** Oct 29, 2021
†**Corresponding Author**
E-mail: songhk@sejong.ac.kr
Tel: +82-2-3408-3890 Fax: +82-2-3409-4264
Won-Seok Lee https://orcid.org/0000-0002-5317-6136
Jun-Yong Jang https://orcid.org/0000-0001-7026-3453
Hyoung-Kyu Song https://orcid.org/0000-0002-3274-4982

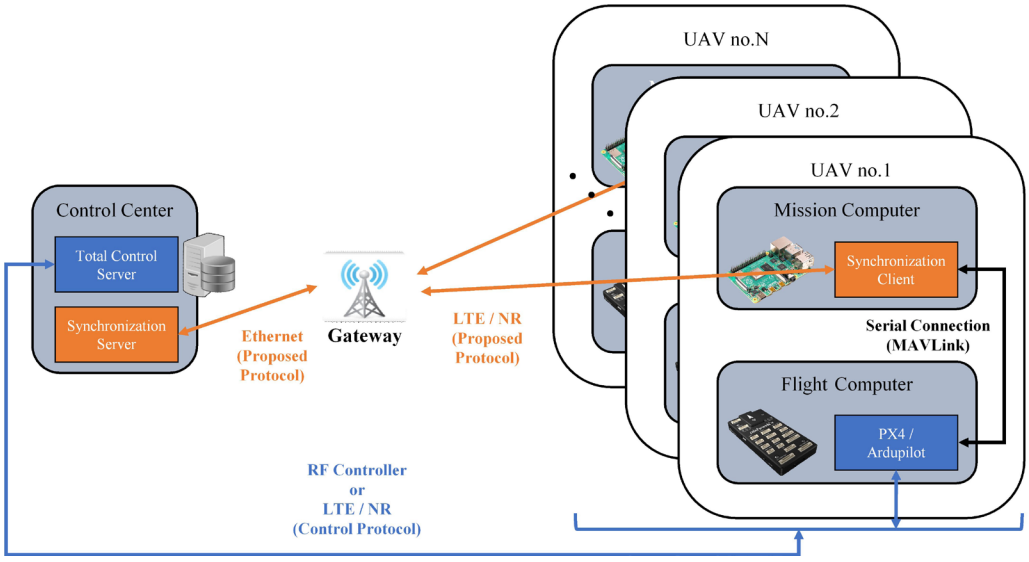


Fig. 1. Overall structure of the proposed system for multiple UAVs.

해 상대적으로 작다. 그렇기 때문에 채널 확보 및 패킷 전달 시간에 대한 불안정성을 효과적으로 줄일 수 있다면, 시간 동기화의 정밀도를 밀리 초 아래로 줄이는 것이 가능하다.

무인비행체와 관련한 연구에서 중요한 주제 중 하나는 다수의 무인비행체를 관제하기 위한 시스템을 설계하는 것이다. 무인비행체는 사고의 위험성이 크고 사고가 날 경우 피해 규모가 클 가능성이 높다. 그렇기 때문에 중앙에서 무인비행체의 상태를 감시하고 통제할 수 있는 관제 시스템이 필수적이다. 시간 동기화 기술은 관제 시스템이 안정적으로 무인비행체를 통제하는데 기반이 되는 핵심 기술 중 하나이다. 특히 군집 비행을 이용한 공연 및 영상 촬영과 같이 관제 시스템을 이용한 응용에서 시간 동기화는 중요한 역할을 한다. 군집 비행을 이용한 공연에서 다수의 무인비행체는 짧은 시간 동안 가까운 거리에서 정밀한 비행 임무를 수행한다. 임무 수행을 위한 기준 시간에 오차가 있다면 충돌로 인한 사고의 위험이 크다. 다수의 무인비행체를 이용한 영상 촬영의 경우 촬영한 영상을 합하여 최종 영상을 만들 때 기준 시간이 다르면 매끄러운 영상을 만들 수 없다. 언급한 응용들과 같이 다수의 무인비행체를 제어하거나 데이터를 통합하는 응용에서 시간 동기화 기술은 핵심적인 역할을 한다.

다수의 무인비행체를 제어하는 네트워크는 무인비행체의 동적 특성과 사고의 위험성으로 인해 다른 무선 네트워크보다 높은 정밀도의 시간 동기화가 필요하다. 기존에 무선 네트워크를 위해 제안된 시간 동기화 기술들은 구성 노드가 베터리와 같이 제한된 에너지를 이용하여 동작하는 상황을 고려했다. 그렇기 때문에 짧은 주기로 패킷 교환을 하는 것이 어렵고 동기화 프로토콜의 구조도 간단할 필요가 있었다. 이러한 제한으로 인해 무인비행체 네트워크에서 기존의 기술들을 이용하여 수 밀리 초 수준의 동기화 성능을 제공하는 것은 어려운 상황이다. 본 논문에서 제안하는 시간 동기화 시스템은 무인비행체 네트워크에서 무인비행체에 탑재되는 모든 임무컴퓨터 뿐만 아니라 비행제어컴퓨터 또한 수 밀리 초 수준의 시간오차 내로 시간을 동기화한다. 이를 위해 정밀한 시간오차 측정을 위한 프로토콜을 이용하며, 시스템의 확장성을 높이기 위해 무인비행체에 탑재된 비행제어컴퓨터가 MAVLink 프로토콜을 이용한다면 어떤 무인비행체라도 시스템에 편입하는 것이 가능하도록 설계했다. 또한 비행제어컴퓨터의 시간 등 내부 변수에 대한 직접적인 변경을 요구하지 않기 때문에 비행제어컴퓨터의 소프트웨어 변경 또한 요구하지 않는다.

2. 동기화 시스템의 구조

본 논문에서 제안하는 시스템은 복수의 서버-클라이언트 구조로 동작하며, 무인비행체의 항법 기능을 제어하고 모니터링하기 위한 관제 시스템과 무인비행체에 탑재된 컴퓨터들의 시간을 동기화하기 위한 시간 동기화 시스템으로 구성된다. 시스템에 속하는 모든 무인비행체는 항법 계산을 위한 비행제어컴퓨터와 임무 수행을 위한 하드웨어 및 소프트웨어가 동작하는 임무컴퓨터가 탑재된다고 가정한다. 임무컴퓨터의 경우는 리눅스 기반의 소형 컴퓨팅 보드를 대상으로 하며, 비행제어컴퓨터의 경우 MAVLink로 통신이 가능한 모든 보드를 대상으로 한다. Fig. 1은 무인비행체의 비행제어컴퓨터로 Pixhawk 보드가 사용되고 임무컴퓨터로 Raspberry Pi 보드가 사용되는 경우를 보인다. 두 보드는 비행제어와 임무를 위한 컴퓨터로 사용되는 대표적인 하드웨어 플랫폼이며, 본 논문에서 대상으로 하는 기준에 부합하는 컴퓨터들이다. Pixhawk와 Raspberry Pi 보드는 유선 serial 연결을 통해 데이터를 주고받을 수 있으며, 데이터 교환을 위한 프로토콜로 MAVLink를 이용한다. MAVLink는 Pixhawk에 설치되는 모든 펌웨어에서 공통으로 사용하는 통신용 프로토콜이며, 비행제어컴퓨터의 항법 정보와 기능을 조회하고 제어하기 위한 메시지 교환 기능을 제공한다.

Fig. 1에서 무인비행체를 관제하는 시스템의 구성요소와 메시지의 상호작용은 파란색으로 표현되어 있다. 각 무인비행체의 소

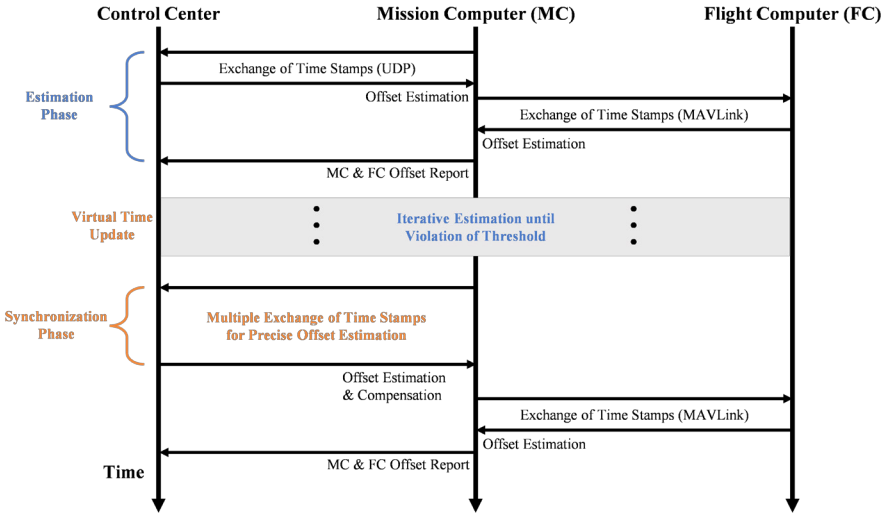


Fig. 2. Process of the time synchronization protocol.

프트웨어는 MAVLink 프로토콜을 이용하여 다양한 방법으로 관제서버와 연결할 수 있다. 무인비행체를 제어하기 위한 방법으로 무선조종기의 RF 통신 방식이 사용될 수 있으며, LTE 혹은 NR 등의 이동통신 인프라를 이용하여 비가시권에서 무인기를 제어하는 것도 가능하다. Fig. 1에서 주황색으로 표현된 구성요소와 상호작용은 시간 동기화 시스템을 나타낸다. 시간 동기화 시스템에서 다수의 클라이언트는 무인비행체의 임무컴퓨터에 설치되며, 관제서버가 동작하는 컴퓨터에 서버가 위치한다. 제안하는 시간 동기화 프로토콜은 각 클라이언트가 설치된 컴퓨터의 시간을 서버가 결정한 기준시간으로 동기화를 진행한다. 클라이언트는 서버가 적절한 기준시간을 결정할 수 있도록 임무 컴퓨터 및 비행제어컴퓨터의 시간정보를 서버로 전송하기 위한 기능이 구현된다. 시간 동기화 시스템의 서버와 클라이언트 간 연결은 이동통신 인프라를 통한 소켓 통신을 이용하여 구현된다. 임무컴퓨터는 이동통신 인프라와의 연결을 위해 이동통신 모뎀을 포함한다. 시간 동기화는 시스템에 구현된 소켓 통신 인프라 위에서 제안 프로토콜을 통해 동작하도록 구현되었다.

3. 시간 동기화 프로토콜

제안하는 시간 동기화 프로토콜은 클라이언트에서 모든 시간 오차를 추정하고 보상하는 과정으로 진행된다. 프로토콜에서 서버의 역할은 오차 추정을 위한 기준시간을 계산하고 클라이언트의 시간 측정 요청에 응답을 하는 것이다. Fig. 2는 시간 동기화 프로토콜의 구조를 보여준다. Fig. 2의 과정은 크게 시간오차 측정 단계와 시간오차 보상 단계로 나눌 수 있다. Fig. 2에서 시간오차 측정 단계에 해당하는 부분은 시간 축에서 파란색으로 표현되어 있으며, 시간오차 보상 단계에 해당하는 부분은 주황색으로 표현되어 있다. 시간오차 측정 단계에서는 임무컴퓨터의 클라이언트가 서버의 기준시간과 임무컴퓨터의 시간 사이의 오차를 측정하기 위해 서버에 측정 값 교환을 요청한다. 오차를 측정하는 과정은 NTP에서 서버와 클라이언트 간 시간오차를 측정하는 방법과 같다. 식 (1-4)는 NTP에서 수행하는 시간오차 측정을 수식으로 나타낸 것이다.

$$\hat{e}_m = \left\{\left(t_2 - t_1\right) - \left(t_4 - t_3\right)\right\} / 2, \quad (1)$$

$$t_2 = t_1 + e_m + d + n_1, \quad (2)$$

$$t_4 = t_3 - e_m + d + n_2, \quad (3)$$

$$\hat{e}_m = e_m + \left(n_1 + n_2\right) / 2 \quad (4)$$

여기서 t_1, t_2, t_3, t_4는 각각 클라이언트에서 패킷을 전송할 때의 시간, 클라이언트의 패킷이 서버에 도착했을 때의 기준시간, 서버에서 응답 패킷을 전송할 때의 기준시간, 서버의 패킷이 클라이언트에 도착했을 때의 시간을 의미한다. 또한 d는 두 노드 사이에서 패킷이 전달되는 동안 걸리는 시간을 의미한다. 식 (1-4)의 과정은 서버와 클라이언트에서 측정한 타임스탬프를 패킷의 형태로 교환하기 때문에 타임스탬프의 획득과 실제 전송 시간 사이에는 오차가 존재하고, 전송 과정에서도 임의의 지연 시간이 발생한다. 식 (2)와 (3)에서 임의의 변동요소들의 합을 잡음인 n_1, n_2로 나타낸다. 두 잡음은 각 전송에서 발생한 잡음을 의미한다. 네트워크의 시간 동기화는 변동요소에 해당하는 잡음들을 어떻게 다루는지가 동기화 프로토콜의 정밀도에 큰 영향을 미친다. 임의의 시간 변동인 잡음의 영향을 계산을 통해 제외하면 각 컴퓨터 간 시간오차를 추정할 수 있다. 식 (1)의 $\hat{e}_m$는 추정된 시간오차를 의미한다.

임무컴퓨터는 관제서버와의 시간오차 측정이 마무리되면 비행제어컴퓨터와의 시간오차 측정을 진행한다. Fig. 3은 Fig. 2에서 표현한 임무컴퓨터와 비행제어컴퓨터 간 시간오차 측정 과정을 더 자세하게 나타낸 것이다. Fig. 3의 과정은 시간오차 측정을 위해 MAVLink 프로토콜의 2번 메시지인 SYSTEM_TIME과 111번 메시지인 TIMESYNC를 이용한다. SYSTEM_TIME 메시지는 두 개의 매개변수를 포함한다. 각각은 time_unix_usec

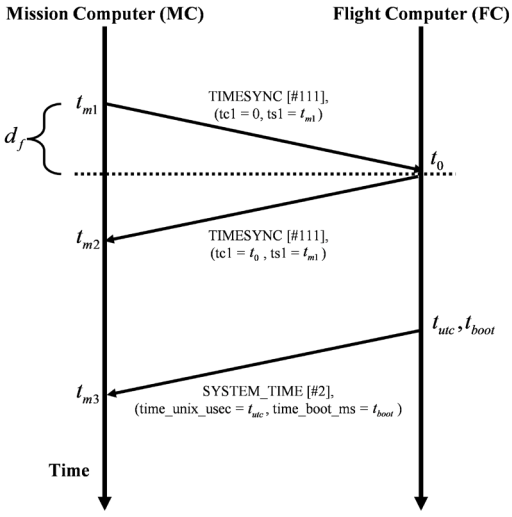


Fig. 3. Interaction between mission and flight computers for estimation of time offset.

와 time_boot_ms로 GPS를 통해 계산된 UTC 시간과 FC가 부팅된 후 경과 시간을 의미한다. SYSTEM_TIME 메시지는 비행제어컴퓨터에서 미리 설정된 시간간격에 따라 반복적으로 전송된다. TIMESYNC 메시지 또한 두개의 매개변수를 포함한다. 각각은 tc1과 ts1으로 보내는 측의 시간은 ts1에 저장되며, 수신 후 응답하는 측은 ts1은 그대로 유지한 상태에서 tc1에 응답하는 측의 시간을 저장하여 전송한다. 응답하는 측에서는 tc1이 0으로 설정된 경우만 응답한다. 비행제어컴퓨터는 마찬가지로 설정된 시간 간격에 따라 TIMESYNC 메시지를 반복하여 임무컴퓨터에 전송한다. 하지만 현재 공개된 최신 버전의 PX4와 Ardupilot 모두 TIMESYNC 메시지를 이용한 시간 동기화 기능은 구현되지 않은 상태이며, TIMESYNC 메시지에 대한 응답 기능만 구현되어 있다. Fig. 3의 과정에서 임무컴퓨터는 메시지 교환을 수행한 t_{m1} , t_{m2} , t_{m3}과 SYSTEM_TIME 메시지에 저장된 UTC 시간을 이용하여 시간오차를 계산한다. 식 (5)와 (6)은 임무컴퓨터에서 수행하는 시간오차 계산을 보인다.

$$d_f = \left(t_{m2} - t_{m1} + n_f\right)/2, \tag{5}$$

$$\hat{e}_f = t_{m3} - \left(t_{utc} + d_f\right). \tag{6}$$

식 (5)는 임무컴퓨터에서 비행제어컴퓨터로 패킷이 전달되는 시간을 계산하는 것이다. TIMESYNC 메시지의 경우는 비행제어컴퓨터로부터 바로 응답 메시지가 오기 때문에 전송한 시간과 응답이 도착한 시간의 중간 값을 전송 지연 시간으로 결정했다. 하지만 실제 패킷이 도착하고, 처리된 후 응답 패킷이 생성되는 시간이 있기 때문에 오차를 고려할 필요가 있다. 식 (5)에서 n_f는 패킷 처리 시간에 대한 오차를 나타낸 것이다. 식 (6)은 계산된 단일 방향의 전송 지연 시간인 d_f를 이용하여 SYSTEM_TIME 메시지가 전달되는 동안 변한 비행제어컴퓨터의 UTC 시간을 계산하고 패킷을 받은 시간과의 차이를 계산한다. 식 (6)에서도 SYSTEM_TIME과 TIMESYNC 메시지는 길이가 다르기 때문에 처리 시간과 전송에 걸리는 시간이 다를 수 있다. 그렇기 때문에 여전히 오차가 존재한다. 임무컴퓨터는 오차를 최소화하기 위해 적절한 주기로 식 (5)의 연산을 수행하면서 d_f를 평균 값으로 갱신한다.

시간오차 보상 단계는 임무컴퓨터의 시간과 관제서버의 시간 사이의 오차가 시스템에 설정된 문턱 값을 넘어가면 진행된다. 제안 시스템은 MAVLink가 동작하는 모든 비행제어컴퓨터를 대상으로 하기 때문에 비행제어컴퓨터에 대한 소프트웨어 변경은 고려하지 않는다. 그렇기 때문에 서버는 기준시간을 적절히 결정하고 임무컴퓨터들이 서버의 기준시간에 동기화를 수행하여 시스템 내의 모든 컴퓨터 간 오차를 최소화한다. 식 (7)은 서버에서 기준시간을 결정하기 위한 방법을 나타낸다.

$$t_s = \arg\min_{t} \sum_{u=1}^{U} \left| t - \left(t_p + e_f\left[u\right]\right)\right|. \tag{7}$$

여기서 t_p와 t_s는 기존의 기준시간과 새로 결정된 기준시간을 의미한다. 또한 U는 시스템에 속하는 전체 무인비행체의 수를 의미하며, $e_f\left[u\right]$는 u번째 무인비행체에서 계산한 임무컴퓨터와 비행제어컴퓨터 간 시간 오차를 의미한다. 기준시간의 변경은 새로운 오차 계산을 요구하기 때문에 식 (7)의 계산 주기를 너무 짧게 하면 시간오차는 수렴하지 않고 큰 변동성을 보인다. 그래서 기준시간의 변경 주기는 임무컴퓨터가 시간오차를 계산하는 주기보다 충분히 길게 설정할 필요가 있다.

임무컴퓨터는 관제 시스템에 충분한 수준의 시간오차를 보장하기 위해 오차 보상 단계에서 정밀한 동기화를 수행해야 한다. 제안 시스템에서는 다수의 무인비행체가 임무를 수행하는데 충분한 시간 동기화 성능을 보장하기 위해 NTP의 시간 오차 측정 방법을 개선한 알고리즘을 이용한다 (Lee et al. 2020). Lee et al. (2020)은 다수의 무인비행체가 포함된 네트워크에서 임무컴퓨터 간의 시간을 동기화하기 위한 시간오차 측정 방법을 제안한다. Lee et al. (2020)이 제안한 시스템은 비행제어컴퓨터의 시간은 고려하지 않지만 임무컴퓨터 간 시간오차는 수 밀리 초의 정밀도로 추정하는 결과를 보인다. 본 논문의 제안 시스템은 임무컴퓨터와 식 (7)에서 계산한 시간 사이의 정밀한 오차를 측정하기 위해 Lee et al. (2020)의 시간오차 측정 방법을 이용한다.

4. 시제품 및 측정 결과

Fig. 4는 제안 시스템의 소프트웨어 구조를 보인다. 마찬가지로 관제 시스템에 속하는 구성요소는 파란색으로 표현되며, 시간 동기화 시스템에 속하는 구성요소는 주황색으로 표현되었다. 관제 시스템은 사물인터넷 국제 표준 플랫폼인 oneM2M 플랫폼 기반의 소프트웨어로 구현되었다 (Choi et al. 2017, Lee et al. 2020). 관제 시스템의 oneM2M 서버는 내부에 데이터 베이스를 포함하며, 트리 구조로 데이터를 관리한다. Fig. 4의 데이터베이스에는 각 무인비행체로부터 전달받은 시간 정보가 저장된다. OneM2M은 국제 표준 플랫폼이기 때문에 표준에 따라 인터페이스만 설계된다면 어떤 소프트웨어라도 데이터 교환이 가능하다.

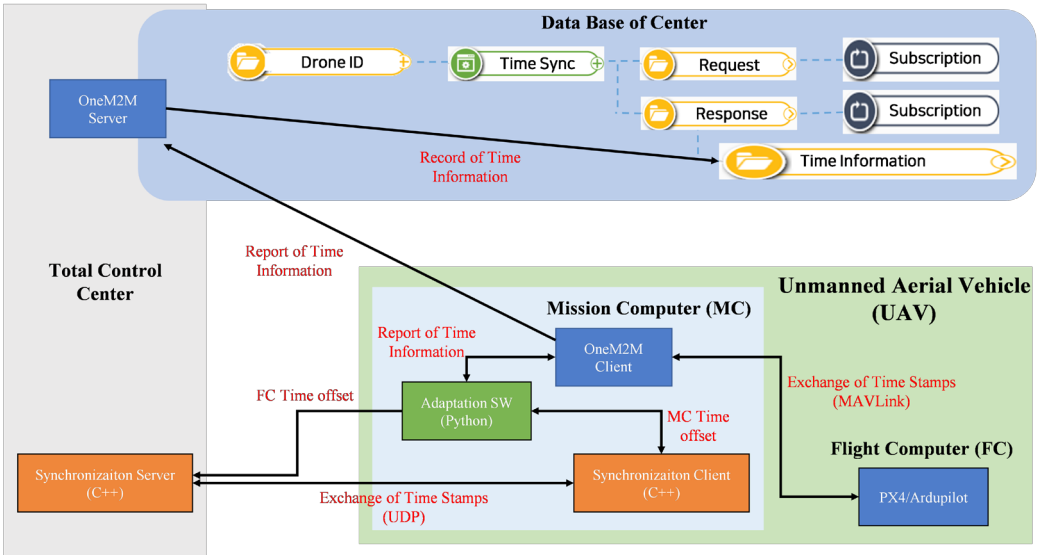


Fig. 4. Software structure of the proposed system.

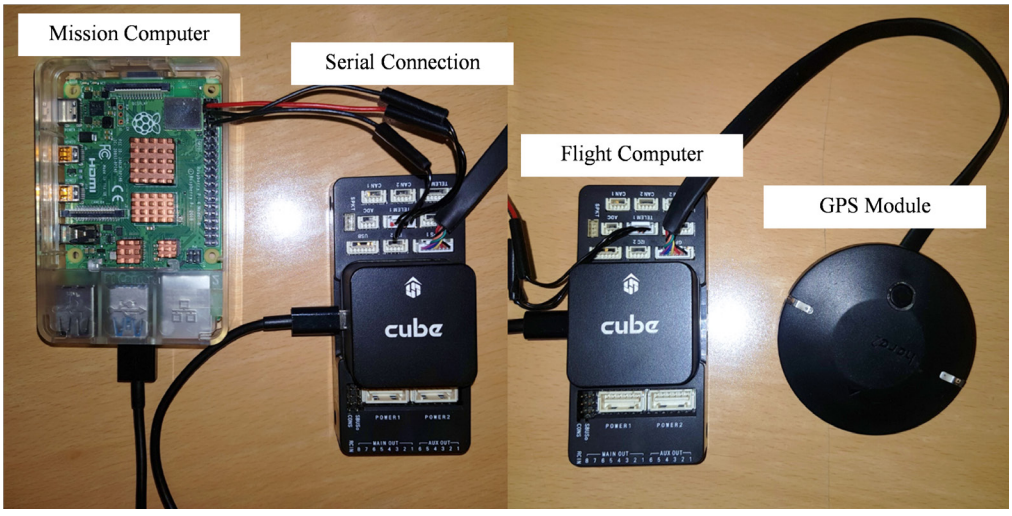


Fig. 5. Mission and flight computers for the prototype.

비행제어컴퓨터는 일반적으로 하나의 serial 연결만을 지원한다. 그렇기 때문에 관제 시스템으로부터 비행제어컴퓨터의 데이터를 넘겨받기 위해 중계 역할을 하는 소프트웨어가 필요하다. Fig. 4의 Adaptation SW가 중계 역할을 수행하는 소프트웨어에 해당한다. Adaptation SW는 oneM2M 클라이언트 그리고 시간 동기화 시스템의 클라이언트와 서버 사이에서 데이터를 중계하기 때문에 각 모듈과 데이터를 교환하기 위한 인터페이스가 구현된다. oneM2M 클라이언트와는 MAVLink 메시지 교환을 중계하여 시간오차 측정을 지원하고, 임무 및 비행제어 컴퓨터들의 시간오차가 측정되면 oneM2M 클라이언트를 통해 서버로 정보를 전송하기 위한 중계를 수행한다. Adaptation SW는 다양한 종류의 인터페이스가 구현되어야 하기 때문에 개발 속도가 빠르고 확장성이 좋은 Python을 이용하여 개발을 진행했다. 시간 동기화 시스템의 서버와 클라이언트는 시간 오차 추정의 정확도를 높이기 위해 동작 속도를 높이는 것이 중요하기 때문에 C++를 개발 언어로 선택했다. Fig. 5는 무인비행체에 탑재하기 위한 하드웨어 구조를 보인다. 비행제어컴퓨터로 Pixhawk Cube를 이용했으며, 임무컴퓨터로 Raspberry Pi 4를 이용했다.

구현 시스템에서는 Pixhawk의 가장 빠른 데이터 샘플링 간격이 10 밀리 초인 것을 고려하여, 중간 값인 5 밀리 초를 시간 동기화의 문턱 값으로 설정했다. Fig. 6는 oneM2M 플랫폼 기반의 인터페이스가 구현된 관제 시스템의 웹 어플리케이션 화면을 보인다. Fig. 6의 화면은 3대의 무인비행체에 대한 측정 결과를 왼쪽부터 시간 순으로 보인다. 각 무인비행체에 대한 측정 결과에서 fc_offset과 mc_offset은 비행제어컴퓨터와 임무컴퓨터의 시간오차를 밀리 초 단위로 보인다. 밀리 초보다 작은 시간 값은 화면에서 생략된다. 결과를 통해 3대의 무인비행체의 시간오차가 5 밀리 초와 가까운 범위에서 수 밀리 초의 값으로 유지되는 것을 확인

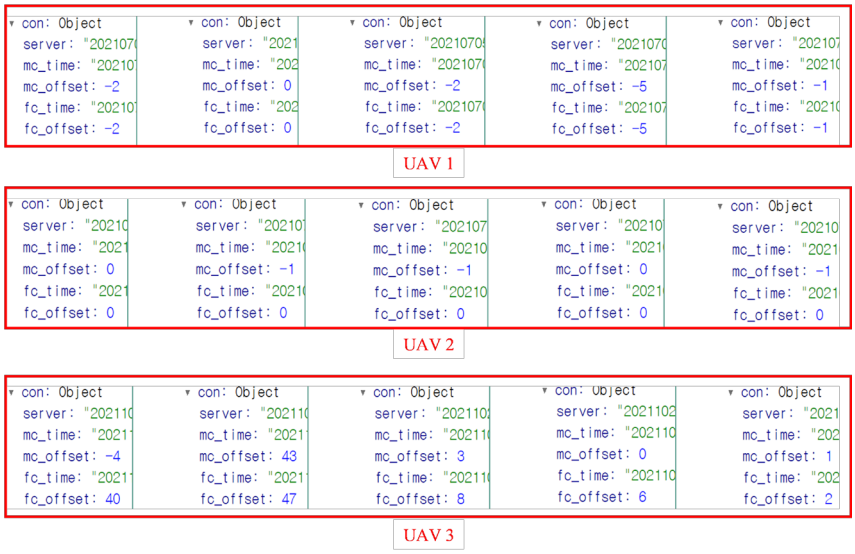


Fig. 6. Estimation and synchroniz ation of the time offset using the prototype.

할 수 있다. 시스템에서 설정한 시간오차의 문턱 값은 5 밀리 초지만 서버의 기준 시간이 식 (7)에 따라 변동하기 때문에 기존에 오차가 잘 맞았던 컴퓨터의 시간오차가 오히려 벌어지는 현상이 관측되기도 한다. 하지만 충분한 시간이 지나면 시간오차의 값은 수렴하게 되고 3대의 무인비행체의 컴퓨터 모두 마지막에는 작은 시간오차가 관측되는 것을 확인할 수 있다.

5. 결론

본 논문은 다수의 무인비행체를 관제하는 시스템에서 무인비행체에 탑재된 모든 컴퓨터들의 시간을 관제컴퓨터의 기준시간으로 동기화하기 위한 시스템을 제안한다. 시간 동기화를 위한 시스템은 복수의 서버-클라이언트 구조로 구성되며, 무인비행체의 임무컴퓨터가 서버의 기준시간과 임무컴퓨터 및 비행제어컴퓨터의 시간 사이의 오차를 측정하고 시스템이 설정한 문턱 값을 넘어가면 동기화를 수행한다. 비행제어컴퓨터에 대한 확장성을 위해 비행제어컴퓨터를 직접 제어하지 않고 기준시간의 변경과 임무컴퓨터의 동기화만으로 시간 동기화를 수행한다. 시제품은 관제 시스템과의 연동 및 확장성을 위해 국제 표준 플랫폼을 이용하여 개발했으며, 성능 측정 결과 시제품의 시간 동기화 시스템이 목표로 설정한 시간오차와 가깝게 임무컴퓨터와 비행제어컴퓨터의 시간을 동기화하는 것을 확인할 수 있다.

ACKNOWLEDGMENTS

본 연구는 국토교통부/국토교통과학기술진흥원, 과학기술정보통신부, 산업통산자원부의 지원으로 수행되었음 (과제번호 1615011800).

AUTHOR CONTRIBUTIONS

Methodology, Jun-Yong Jang; software, Jun-Yong Jang, Won-Seok Lee; validation, Jun-Yong Jang, Won-Seok Lee and Hyoung-Kyu Song; writing—original draft preparation, Won-Seok Lee; writing—review and editing, Won-Seok Lee, Hyoung-Kyu Song; supervision, Hyoung-Kyu Song; project administration, Hyoung-Kyu Song.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

Choi, S., Sung, N., Park, J., Ahn, I., & Kim, J. 2017, Enabling drone as a service: OneM2M-based UAV/drone management system, in 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Milan, Italy, 4-7 Jul. 2017. https://doi.org/10.1109/ICUFN.2017.7993739

Elson, J., Girod, L., & Estrin, D. 2002, Fine-grained network time synchronization using reference broadcasts, ACM SIGOPS Operating Systems Review, 36, 147-163. https://doi.org/10.1145/844128.844143

Hasan, M. K., Saeed, R. A., Hashim, A. H. A., Islam, S., Alsaqour, R. A., et al. 2012, Femtocell network time synchronization protocols and schemes, Research Journal of Applied Sciences, Engineering and Technology, 4, 5136-5143

Johannessen, S. 2004, Time synchronization in a local area network, IEEE control systems Magazine, 24, 61-69. https://doi.org/10.1109/MCS.2004.1275432

Lasassmeh, S. M. & Conrad, J. M. 2010, Time synchronization in wireless sensor networks: A survey, In Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon), Concord, NC, USA, 18-21 Mar. 2010. https://doi.org/10.1109/SECON.2010.5453878

Lee, W. S., Jang, J. Y., & Song, H. K. 2020, Time Management System for Applications of UAV Network, Journal of Positioning, Navigation, and Timing, 9, 405-409. https://doi.org/10.11003/JPNT.2020.9.4.405

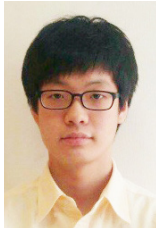
Li, P., Gong, H., Peng, J., & Zhu, X. 2019, Time Synchronization of White Rabbit Network Based on Kalman Filter, In 2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE), Xiamen, China, 18-20 Oct. 2019. https://doi.org/10.1109/EITCE47263.2019.9094978

Mills, D. L. 1991, Internet time synchronization: the network time protocol, IEEE Transactions on communications, 39, 1482-1493. https://doi.org/10.1109/26.103043

Mills, D., Martin, J., Burbank, J., & Kasch, W. 2010, Network time protocol version 4: Protocol and algorithms specification. https://www.hjp.at/doc/rfc/rfc5905.html

Solis, R., Borkar, V. S., & Kumar, P. R. 2006, A new distributed time synchronization protocol for multihop wireless networks, In Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, USA, 13-15 Dec. 2006. https://doi.org/10.1109/CDC.2006.377675

van Greunen, J. & Rabaey, J. 2003, Lightweight time synchronization for sensor networks, In Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, San Diego, CA, USA,19 Sep. 2003, pp.11-19. https://doi.org/10.1145/941350.941353



Won-Seok Lee received the B.S. and M.S. degrees in Information and Communication Engineering, Sejong University, Seoul, Korea, in 2016 and 2018, respectively. He is currently working toward to the Ph.D. degree in the Department of information and communications engineering, Sejong University, Seoul, Korea. His research interests are in the areas of signal processing for wireless communication system and ambient RF communication system.

Jun-Yong Jang received the B.S. degree in the Department of Information & Communication Engineering, Sejong University, Seoul, South Korea, in 2020. He is currently pursuing the M.S. degree in the Department of Information & Communications Engineering, Sejong University, Seoul, South Korea. His research interests are in the area of wireless communication systems design and MIMO signal processing.

Hyoung-Kyu Song was born in Chung Cheong-Bukdo, Korea on May 14 in 1967. He received B.S., M.S., and Ph.D. degrees in electronic engineering from Yonsei University, Seoul, Korea, in 1990, 1992, and 1996, respectively. From 1996 to 2000 he had been managerial engineer in Korea Electronics Technology Institute (KETI), Korea. Since 2000 he has been a professor of the Department of information and communications engineering, Sejong University, Seoul, Korea. His research interests include digital and data communications, information theory and their applications with an emphasis on mobile communications. The corresponding author of this paper.