

원격운용 시스템의 네트워크 성능분석을 위한 시간동기화 방안에 관한 연구

A Study on Time Synchronization Method for Analyzing the Network Performance of Remote Control System

양 동 원[★], 김 남 곤^{*}, 김 도 종^{*}

DongWon Yang[★], Namgon Kim^{*}, Dojong Kim^{*}

Abstract

With the development of artificial intelligence and unmanned technologies, the remote surveillance/autonomous driving systems have been actively researched. For an effective performance analysis of the developed remote control system, it is important to record the data of it in real time. In addition, in order to analyze the performance between the control system and the remote system, the recorded data from them should be synchronized with time. In this paper we proposed a novel time synchronization method for the remote control system. The proposed remote control system satisfies the time difference of the recorded data within 1 ms, and we can reduce the time difference by using a CPU shielding and affinity setting. The performance of the proposed method was proved through various network data storage experiments. And the experiments confirmed that the proposed method can be applied to recording devices of unmanned ground vehicles and control vehicles. The proposed method will be used as a method for analyzing network data of UGV-R (Unmanned Ground Vehicle - Reconnaissance).

요 약

인공지능 및 무인 감시, 자율화 기술의 발전으로 인해 무인으로 운용되는 원격 감시/자율 주행 시스템의 개발이 활발히 연구되고 있다. 개발되는 원격운용 및 제어 시스템의 효과적인 성능분석을 위해서는 원격운용 시스템의 데이터를 실시간으로 기록하고 그 데이터를 분석하는 일이 중요하다. 또한, 통제 시스템과 원격 시스템 간의 성능분석을 위해서는 각 시스템의 기록데이터 간 시간동기화가 필수적으로 요구된다. 본 논문에서는 원격운용 시스템의 기록데이터 분석을 위한 GPS 기반 시간동기화 설계 방안을 제시한다. 제안 방법은 기록데이터에 GPS 신호를 활용한 정밀한 시간정보를 기록함으로써 원격 시스템이 기록데이터 시간 오차를 1ms 이내로 만족할 수 있도록 하였으며, OS 내 설정을 통해 이더넷 통신을 특정 CPU가 전담하도록 하는 CPU shielding 및 affinity 설정 기법을 통해 시간동기화 성능을 개선하였다. 제안 방법은 단계별 실험 및 네트워크 데이터 저장 실험을 통해서 그 성능을 입증하였으며, 무인수색차량의 무인차량과 통제차량 기록장치에 적용할 수 있음을 확인하였다. 제안 방법은 향후 무인수색차량의 네트워크 데이터 분석 방법으로 활용될 예정이며, 활용 중에 발생하는 다양한 분석을 통해 성능 개선을 해 나갈 예정이다.

Key words : CPU shielding and affinity, Network Data Analysis, Network Data Recording, Remote Control System, Time Synchronization Method

* Ground Technology Research Institute(GTRI), Agency for Defense Development(ADD), Republic of Korea

★ Corresponding author

E-mail : dwyang@add.re.kr, Tel : +82-42-821-3146

Manuscript received Feb. 2, 2022; revised Mar. 20, 2022; accepted May. 17, 2022.

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

I. 서론

인공지능 기술의 발달과 영상처리 기술, 고속무선통신 기술의 발달에 힘입어 무인화 기술은 급속도로 발전하고 있다. 이에 따라 국방 분야에서도 AI 기술이 접목된 무인체계 및 그와 관련된 기술 개발이 많이 진행되고 있다 [1-3]. 이런 기술 개발 중 하나로 감시정찰의 목적을 갖는 무인차량에 대한 개발이 진행되고 있으며, 운용병은 후방에 떨어져서 원격으로 무인차량을 제어하는 역할을 수행하게 된다. 원격제어를 통해 임무를 수행하는 무인차량은 이동, 주행, 감시, 사격 등의 임무를 원격으로 수행하게 되며 자율주행 기술을 활용하여 원격제어 없이 자율로 주행하기도 한다. 이러한 기술은 모두 인공지능 기술과 고속의 실시간 영상처리 기술 및 무선통신 기술의 결합으로 구현되게 된다.

그러나, 본 기술을 효과적으로 개발하기 위해서는 유무선이 결합된 원격운용 시스템의 네트워크 신호를 분석하여 설계대로 구현되었는지 검증하는 것이 필요하다. 수십 대 이상의 다양한 컴퓨터가 집적된 무인차량은 각 컴퓨터 간에 이더넷 통신을 기반으로 메시지를 주고받고 있으며, 무인차량과 원격 운용시스템 간에는 고속 무선 통신을 기반으로 제어신호를 주고받고 있다[4]. 고속의 자율주행과 감시, 사격 등의 다양한 임무처리를 무인차량이 효과적으로 수행하기 위해서는 다양한 컴퓨터 간에 유기적인 메시지 전달이 필요하며, 이를 설계에 반영하고 실제 설계대로 메시지를 송수신하는 것을 확인하는 것이 중요하다.

일반적으로 유선망으로 연결된 시스템의 네트워크 데이터를 분석하는 기법은 많이 연구됐다[5-7]. 네트워크 기반의 다양한 시간동기화 방법(NTP: Network Time Protocol, PTP: Precision Time Protocol)을 활용하여 각 컴퓨터 내부의 시간을 하나로 동기화한 뒤 메시지의 순서를 기록하는 방법을 통해서 분석이 이루어졌으며, 대형 네트워크 서버를 비롯한 금융망 시스템 등에 다양하게 적용되어 그 기술의 검증이 이루어졌다. 또한, 무선망으로 연결된 장치들은 장치 간의 네트워크 메시지 분석보다는 장치 간 시간동기화 기법이 많이 연구되었는데 일반적으로 GPS(Global Positioning System) 신호를 기반으로 각 장치의 시계를 동기화하는 방법으로 구현되었다[8-10]. 본 논문에서는 원격 운용시스템과 무인차량으로 구성된 유무선 복합 시스템의 네트워크 데이터 분석을 위한 시간동기화 방법을 새롭게 제안했으며, 제안된 방법의 시간동기화 정확도를 측정하여 분석한 결과

를 나타내었다. 제안하는 방법은 GPS 신호를 기반으로 원격으로 수Km 이상 떨어진 시스템의 시간을 동기화하였으며, 각 운용 장치에 있는 데이터 기록장치는 모든 이더넷 데이터의 시간을 tagging 하여 기록함으로써 메시지 분석이 가능하도록 설계하였다. 또한, 데이터 기록장치에는 실시간 OS(RedHawk)와 네트워크 송수신 전담 core 및 IRQ(Interrupt ReQuest)를 지정하여 적용함으로써 네트워킹 메시지 수신 간 지연을 최소화하였다. 제안하는 방법은 GPS 신호의 차이부터 각 기록장치에 저장되는 시간의 차이까지 단계적으로 분석을 수행하여 제안하는 방법이 유무선 망이 혼합된 시스템에 효과적으로 적용할 수 있음을 입증하였다.

본 논문은 다음과 같이 구성된다. 본론 1장에서는 제안하는 기록데이터 시간동기화 방법 및 CPU shielding/IRQ setting 방법에 관하여 기술하였으며, 2장에서는 제안하는 방법의 단계적 검증 실험 결과를 분석하여 나타내었다. 마지막으로 결론에서는 결론 및 향후 개선 방향에 대하여 제안하였다.

II. 본론

1. 제안하는 시간동기화 방안

본 절에서는 제안하는 시간동기화 방법과 실시간 OS 및 네트워킹 SW 설정 방법에 관하여 기술하였다. 이를 위해 먼저 제안 방법을 적용할 수 있는 원격운용 시스템 및 기록데이터 운용 방법에 대한 설명을 가함에 나타내었다. 제안하는 시간동기화 방법에 대한 설명과 실시간 OS 및 네트워킹 SW 적용 방안은 각각 나항과 다항에 나타내었다.

가. 원격운용 시스템 및 데이터 운용 방법

무인으로 운용되는 시스템은 원격지에서 운용되는 무인차량과 그 장비를 통제하는 통제차량으로 구성되며 그림 1과 같다.



Fig. 1. An example of remote control system with network recorder.

그림 1. 시간 동기화된 네트워크 기록장치를 보유한 원격운용 시스템 구성도 예

통제차량은 무선통신을 활용하여 무인차량을 원격제어 할 수 있으며 무인차량은 통제차량의 명령에 따라 원격 주행/자율주행/감시정찰 등의 다양한 임무를 수행하게 된다. 제안하는 방법이 적용된 시스템 구성에는 통제차량과 무인차량 각각 기록장치를 보유하고 있으며 기록장치에는 네트워크로 송수신되는 모든 데이터가 기록된다. 이때, 기록장치에서는 각 시스템에 있는 GPS 신호를 기반으로 생성한 시간정보를 네트워크 데이터와 함께 저장하게 된다. 무인차량과 통제차량의 내부망 구성은 그림 2와 같다. 각 차량 내부 구성품은 내부 구성품 간의 통신을 데이터망을 기반으로 수행한다. 통제차량과 무인차량 간의 통신은 무선통신으로 구성되어 있으며, 각 차량은 별도의 기록기를 가지고 있다. 각 기록기는 각 차량에 설치된 GPS 신호를 이용하여 시간 데이터를 자체적으로 계산하며, 모든 데이터망에 송수신되는 데이터를 각 구성품이 복사하여 기록망으로 한 번 더 송신함으로써 모든 데이터 송수신 결과를 기록장치에서 수신한 뒤 시간정보를 붙여서 저장할 수 있게 된다. 그 결과 무인차량과 통제차량의 각 구성품은 자신이 송수신하는 네트워크 데이터를 기록장치에 시간 순서대로 저장할 수 있으며, 이는 추후 각 차량의 구성품 성능분석에 활용할 수 있다. 또한, 무인차량 기록장치와 통제차량 기록장치 간의 시간동기화를 통해서 무선통신으로 연결된 두 차량 간의 메시지를 시간에 따라 분석할 수 있게 된다. 자세한 시간 동기화 방법은 나항에 나타내었다.

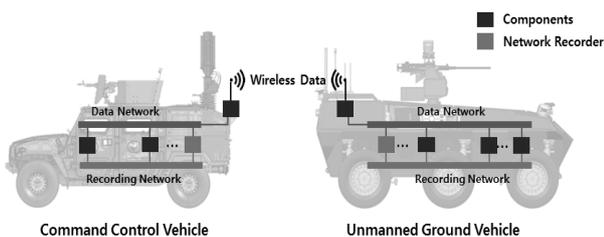


Fig. 2. System network diagram using dual(data and recording) networks and network recorder.
그림 2. 이중망(데이터망, 기록망)과 기록장치를 보유한 망 구성도

나. 시간동기화 방법

무인차량 기록장치에 저장된 네트워크 데이터와 통제차량 기록장치에 저장된 네트워크 데이터를 같은 시간 선상에서 분석하려면 두 기록장치 간 시간을 동기화하는 일이 필요하다. 각각의 기록장치는 GPS 신호를 이용하여 시간정보를 획득하고, 각 기록장치 내부의 clock을 이용하여 세부 시간을 계산하게 된다. 일반적으로 유선

으로 모든 구성품이 연결된 시스템의 경우 시간동기화 서버가 다른 구성품(클라이언트)의 시간을 주기적으로 조사해서 특정 오차 이내로 동기화하는 방법을 사용한다 [5-7]. 그러나, 원격운용 시스템은 네트워크의 연결이 차량 내부에서는 유선으로 구성되지만, 무인차량과 통제차량 간에는 무선통신 구간이 포함되어 있어서 유선에서 적용하는 일반적인 시간동기화 방법의 적용이 어렵다. 따라서 본 논문에서는 다음 그림 3과 같은 시간동기화 방법을 제안한다.

제안하는 시간동기화 방법은 GPS 신호를 기반으로 하고 있다. 위성으로부터 받은 시간정보를 기준으로 기록장치는 GPS 수신부에서 1PPS(Pulse Per Second) 신호와 TOD(Time of Day) 정보를 수신하게 된다. 수신받은 시간은 기록장치 내 SBC(Single Board Computer)에 입력되며 초 단위의 시간인 TOD 정보를 ns 단위로 세분화한다. 세분화 된 시간정보를 활용하여 네트워크 기록데이터가 도착하면 그 데이터에 time stamp를 ns 단위로 기록하여 저장하게 된다. 이때, 무인차량과 통제차량은 원격지에 떨어져서 운용되지만 같은 하드웨어의 기록장치와 GPS 안테나, 같은 SBC를 이용하고 같은 시간관리 SW를 적용함으로써 통제차량 기록장치와 무인차량 기록장치의 기록시간 오차가 1ms 이내가 되도록 관리하였다. 제안하는 원격운용 시스템은 100Hz 이상의 제어신호를 갖는 시스템이기 때문에 1ms 이내의 시간동기화 오차를 갖는다면 무인차량과 통제차량의 제어신호를 시간순서에 따라 분석하는데 문제가 없음을 알 수 있다.

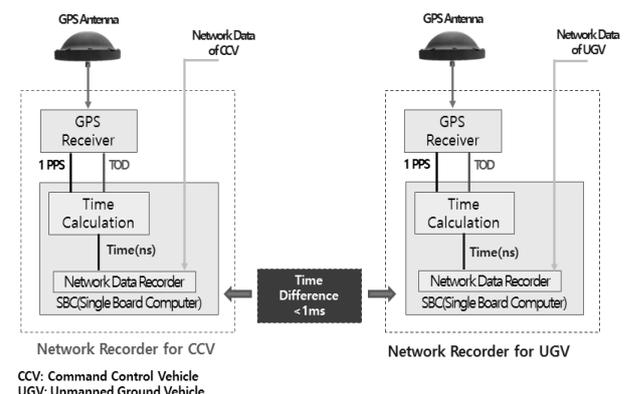


Fig. 3. Block diagram of network recorder using time synchronization method.

그림 3. 시간동기화 방법이 적용된 네트워크 기록장치 구성도

내부 clock을 이용하여 ns 단위의 세부 시간정보를 획득하는 방법은 그림 4와 같다.

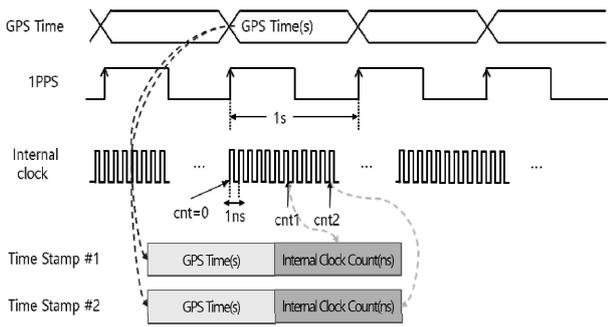


Fig. 4. Time stamp generation method and examples. 그림 4. 타임스탬프 생성 방법 및 예시

초 단위로 들어오는 GPS 시간데이터를 1PPS에 동기화해서 수신한다. 그 후 1초에 한 번씩 발생하는 PPS 신호를 이용하여 내부 clock의 counter를 reset 해주고 저장해야 하는 네트워크 데이터가 수신되면 수신 즉시 그때의 counter 값을 이용하여 그때의 세부 시간(ns 단위)을 기록한 뒤 네트워크 데이터 저장 시 time stamp를 함께 저장하게 된다. 이는 저장 동작으로 인한 지연이 데이터 기록에 반영되지 않도록 하기 위함이며, 데이터 수신과 동시에 time stamp를 생성한 뒤 저장은 나중에 함으로써 저장동작과 관련된 지연으로 인한 시간 계산 오차를 줄일 수 있다. 이 방법을 통해서 동시에 도착하는 여러 개의 데이터의 time stamp 생성 지연을 최소화할 수 있다. 그림 4에는 두 개의 데이터가 각각 cnt1, cnt2의 시점에 도착하였을 때 time stamp #1과 #2가 생성된 예를 나타내었다. 각 time stamp는 t_{s1} 과 t_{s2} 로 다음의 수식으로 계산할 수 있다. 여기서 t_{GPS} 와 t_{tick} 은 위성에서 수신한 초 단위의 시간 값과 내부 clock의 tick 시간을 의미한다.

$$t_{s1} = t_{GPS} + cnt1 \times t_{tick} \quad (1)$$

$$t_{s2} = t_{GPS} + cnt2 \times t_{tick} \quad (2)$$

내부 clock tick은 적용하는 시스템에 맞게 설정 및 변경을 할 수 있으며, 본 논문에서는 ns 단위의 clock tick을 사용하였다. 이는 적용하는 장비의 시간 오차 단위를 1ms 이내로 설계하였기 때문에 세부 저장 시간을 ns 단위로 저장하고 통제차량과 무인차량의 시간 저장 오차는 평균 30us 수준이며, 최대 250us 수준임을 실험을 통해 확인함으로써 제안하는 시간동기화 방법이 무인차량과 통제차량의 시간동기화 방법으로 효과적으로 사용할 수 있음을 확인하였다. 상세한 분석 결과는 2절의 실험 결과에 나타내었다.

다. 실시간 OS 및 네트워킹 SW 적용 방안

기가비트 이더넷 기반의 통신은 실시간성보다 안정적인 전송에 중점을 두고 있는 통신방법이기 때문에 네트워크 데이터의 전송속도는 일정하지 않으며 동일한 데이터를 반복 전송하더라도 전송되는 시간의 차이가 존재한다. 이는 제안하는 방법에서 무인차량과 통제차량의 이더넷 데이터 분석할 때 데이터가 도착하는 시간이 매번 변화할 수 있음을 나타낸다. 즉, 분석을 위해 획득한 데이터의 시간정보가 안정성이 떨어지는 결과를 낳게 된다. 따라서 본 논문에서는 SBC(Single Board Computer)내의 이더넷 통신 기능 구현 시 RedHawk Linux 기반의 CPU shielding 기법과 네트워크 인터럽트를 특정 CPU로 전담케 하는 설정방법[11]을 적용하여 시간동기화 방법의 안정성을 개선하였다.

RedHawk은 RedHat Enterprise 혹은 CentOS와 호환되며 time critical 및 hard real-time 환경에 적합하게 개선된 운영체제이다. 제안 방법에서 적용한 RedHawk Linux는 7.5.x 버전이며, 특정 프로세스(이더넷 통신)가 정해진 CPU core를 활용할 수 있도록 하는 shielding 기능을 적용하였다. 또한, 네트워크 송수신으로 발생하는 인터럽트를 특정 CPU 한 개에서 받을 수 있도록 CPU affinity 설정을 변경함으로써 프로세스 간 인터럽트 교

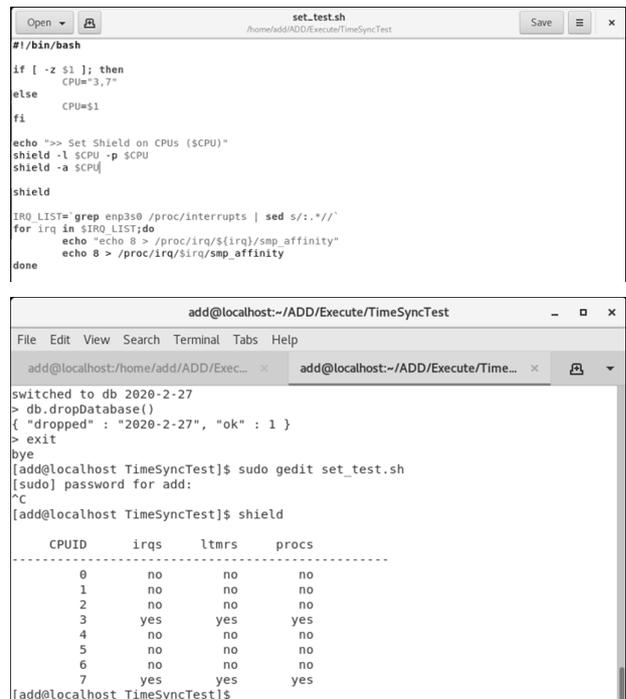


Fig. 5. Script for cpu shielding/affinity setting and the results.

그림 5. RedHawk내 shielding/CPU affinity 설정 script 및 결과 화면

차로 인한 손실을 최소화하였다. 그림 5에는 RedHawk 내 shielding 및 CPU affinity 설정 script의 예와 shielding 설정 결과 화면을 예시로 나타내었다.

제안하는 CPU shielding 및 CPU affinity 설정 방법을 통해 네트워크 데이터 간의 time difference latency값이 평균 64.94us에서 26.93us로 개선되었으며, maximum값이 378us에서 247us로 개선된 결과를 확인하였다. 상세한 실험 및 분석결과는 본문 2-나항에 기술하였다. 이를 통해서 CPU shielding 및 affinity 설정방법이 시간동기화 기능 구현에 필요함을 알 수 있었으며 제안하는 네트워크 데이터 분석 방법에 필요함을 알 수 있었다.

2. 실험 결과

본 장에서는 제안하는 시간동기화 방법을 검증하기 위해 수행한 실험 및 결과를 분석하여 기술하였다. 먼저 시간동기화 방법의 사전 검증방법으로 GPS 수신보드에서 출력되는 1PPS 신호와 TOD 신호를 분석하였으며(가항), 무인차량과 통제차량 기록기의 데이터 저장 실험을 수행하여 그 결과를 분석하였다(나항). 이때 일반 노트북을 이용한 실험결과와 SBC와 RedHawk Linux 및 네트워크 SW/CPU shielding 기법을 적용 여부에 대한 실험결과를 분석하여 최적의 네트워킹 지연시간을 확보할 수 있는 시간동기화 방안을 도출할 수 있었다. 마지막으로 제안하는 방법의 객관적 유용성을 검토하기 위해서 실 환경에서 네트워크 Tap 장비와 성능 비교를 수행하였다(다항).

가. 시간동기화 사전 검증 실험 결과

시간동기화 성능을 사전 검증하기 위해서 각 기록장치에 있는 GPS 수신모듈(GPS Receiver)에 수신되는 1PPS 신호와 TOD 신호가 기록장치 간 오차가 얼마나 발생할 수 있는지 분석하였다. 실험을 위해 아래 그림과 같이 장비를 구성하였다.

두 개의 기록장치를 사용하는 것을 가정해서 두 개의 GPS 수신모듈을 사용하였다. 각각의 수신부는 1PPS 신호와 TOD 신호를 위성신호를 이용하여 생성하고 각각의 신호는 하나의 오실로스코프에 입력하여 파형을 측정하고 그 시간차를 분석했다. 각각 32회의 측정결과를 표 1에 나타내었으며, 결과 파형의 예를 그림 7에 나타내었다. 측정결과 1PPS 신호는 별도의 GPS 안테나를 이용하여 수신하더라도 평균 37ns 수준의 시간차를 보임을 확인할 수 있었으며 시간정보 신호인 TOD 신호는 약

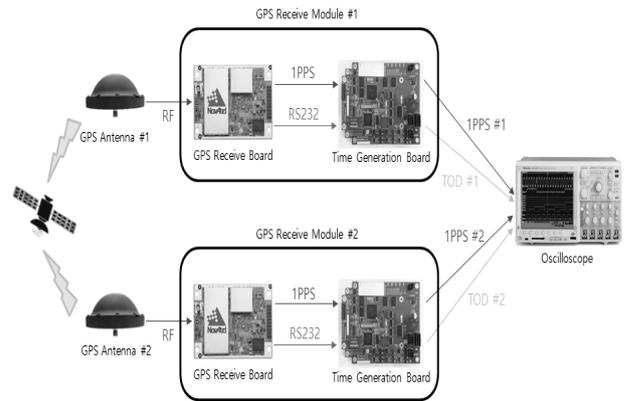


Fig. 6. GPS Receive Module Test Set-up.

그림 6. GPS수신모듈 사전 점검 실험 환경

2.4us의 평균 시간차이를 나타냄을 확인할 수 있었다. 이것은 현재 시간동기화 목표로 설정한 1ms에 비해서 매우 작은 양으로 두 개의 기록장치 간 GPS 신호 및 시간정보 신호는 시간동기화 오차에 크게 영향을 주지 않음을 확인할 수 있었다.

나. 데이터 저장 실험 결과 비교 분석

본 절에서는 기록장치 내 저장된 데이터의 시간동기화 성능을 분석하기 위하여 세 가지 실험을 수행하여 결과를 분석하였다. 시간동기화 성능을 높이는 방법을 단계적으로 분석하기 위해서 먼저 일반 노트북 및 Ubuntu Linux를 활용한 저장장치를 활용한 실험을 먼저 수행하였다.

Table 1. Time Difference Analysis Results for 1 PPS and TOD.

표 1. 1 PPS 및 TOD 신호의 시간차 분석 결과

Time Difference	Avg. (ns)	Min. (ns)	Max. (ns)	Std. (ns)	counts
1PPS	37.94	0.03	233.60	73.04	32
TOD	2402.84	0.28	6791.00	1616.50	32

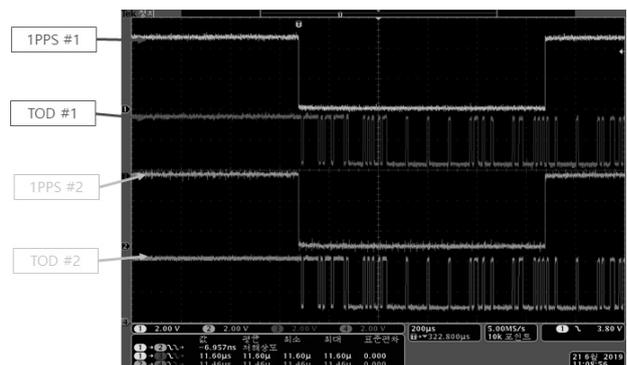


Fig. 7. Experimental Result for 1PPS and TOD signals.

그림 7. 1PPS 및 TOD 신호 측정 파형

그 후 제작된 별도의 기록장치(SBC 활용 및 RedHawk 이용)를 활용하여 두 번째 실험을 수행하였으며, 마지막으로 제안하는 방법과 같은 CPU shielding 방법과 affinity 설정방법까지 적용한 기록장치(SBC, RedHawk)를 활용한 실험결과를 나타내었다.

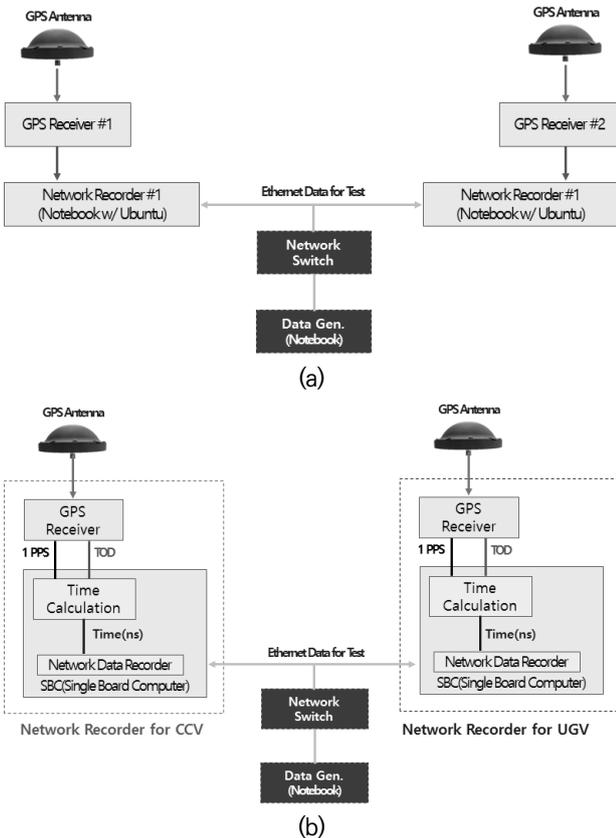


Fig. 8. Time Synchronization Test Environment Set-up. (a) Using Notebook and Ubuntu Linux for Data Recorder. (b) Using SBC and RedHawk Linux for Data Recorder.

그림 8. 시간동기화 실험 환경 세팅

- (a) 노트북과 Ubuntu Linux를 데이터 저장장치로 사용한 환경
- (b) SBC와 RedHawk Linux를 데이터 저장장치로 사용한 환경

그림 8에는 노트북(Ubuntu Linux)를 활용한 실험 세팅과 기록장치(SBC, RedHawk)를 활용한 세팅을 각각 나타내었다. 제안하는 방법은 기록장치를 활용한 세팅방법에 본문 1-다항의 설정방법까지 SW로 적용한 방법에 해당한다. 각 실험에서는 네트워크 스위치를 활용해 두 개의 저장장치로 동시에 전송된 이더넷 데이터를 저장할 때 저장시간 정보를 비교함으로써 시간동기화 오차를 확인하였다.

먼저 그림 8(a) 환경으로 세팅한 일반 노트북과 Ubuntu Linux를 활용한 실험결과를 표 2에 나타내었다. 다양한 데이터 전송속도(10Hz~100Hz)에 대하여 테스트를 수행한 결과 저장시간의 차이가 평균 351.8~549.6 us이며 최대 5.9ms의 시간차이를 나타내어 시간동기화 목표(1.0ms 이내)를 만족할 수 없음을 확인할 수 있었다. 일반 이더넷 데이터의 지연시간으로는 저장시간을 동기화 하는데 어려움이 있음을 확인할 수 있었다.

Table 2. Time Synchronization Performance Result #1-Notebook(Ubuntu).

표 2. 시간동기화 성능분석 결과 #1-Notebook(Ubuntu)

Data Rate(Hz)	Avg. (us)	Max. (us)	Std. (us)	counts
10	351.8	1100.5	239.4	59919
50	549.6	2899.2	118.3	59917
100	526.9	5899.4	439.3	59917

Table 3. Time Synchronization Performance Result #2-SBC(RedHawk).

표 3. 시간동기화 성능분석 결과 #2-SBC(RedHawk)

Data Rate(Hz)	Avg. (us)	Max. (us)	Std. (us)	counts
10	43.26	320.00	36.49	49928
50	73.72	398.82	49.43	49909
100	71.05	367.59	48.28	49923

두 번째로 기록장치를 제작하고 SBC와 Red Hawk Linux를 이용하여 실험한 결과를 표 3에 나타내었다. 수행결과 표 2의 실험결과에 비해서 평균 43~74us 수준 및 최대 400us 수준으로 성능이 개선되었음을 확인할 수 있었다. 시간동기화 목표인 1ms 수준으로 적용할 수 있음을 확인할 수 있었다. 시간동기화 성능분석 결과 #1과 #2를 비교해보면 각 데이터 주기에 따라 SBC(RedHawk)의 경우가 평균이 대폭 감소했음을 그림 9(a)에서 확인할 수 있다.

본문 1-다항에서 논의된 CPU shielding 및 affinity 설정의 효과를 분석하기 위해서 설정 여부에 따른 성능 비교를 추가로 수행하여 표 4에 나타내었다. CPU 설정 결과 시간동기화 오차의 평균값이 30us 이내로 감소함을 확인할 수 있었으며, 최대 시간오차도 250us 이내로 줄어드는 것을 확인할 수 있었다. CPU shielding과 affinity 설정 적용 시 평균시간과 최대값 모두 개선됨을 확인할 수 있었다. 따라서 제안하는 시간동기화 방법으

로 적용하면 1ms 이내의 시간동기화를 만족하는 데 문제가 없음을 확인하였고, 이를 활용해서 유무선 데이터가 복합적으로 있는 무인차량과 통제차량의 인터넷 데이터를 효과적으로 분석이 가능할 것으로 예상된다.

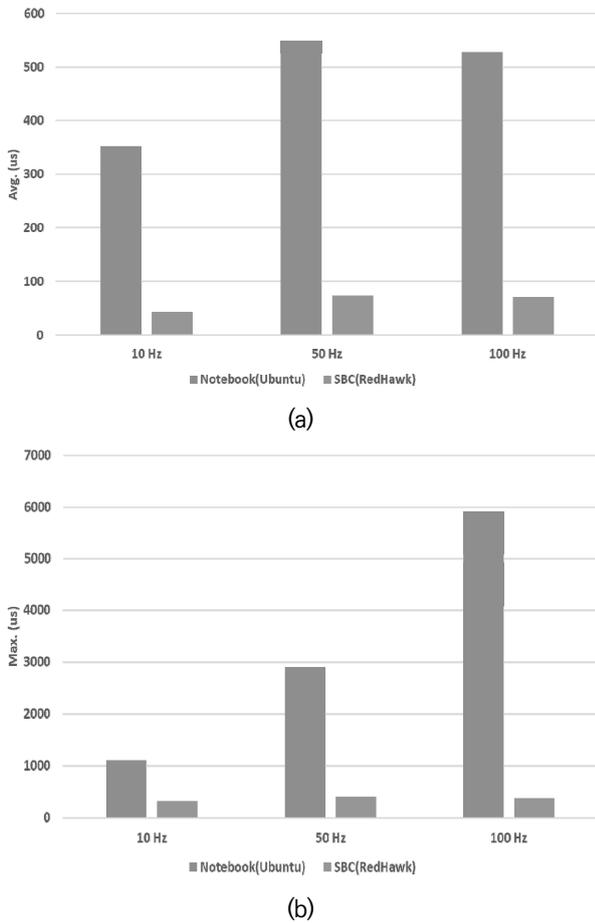


Fig. 9. Comparison Results between Notebook(Ubuntu) and SBC(RedHawk) cases.

- (a) Average time difference.
- (b) Maximum time difference.

그림 9. Notebook(Ubuntu) 세팅과 SBC(RedHawk)세팅의 실험 결과 비교

- (a) 저장 시간차이의 평균 값 (b) 저장 시간차이의 최대 값

Table 4. Time Synchronization Performance Result #3-CPU shielding effect analysis.

표 4. 시간동기화 성능분석 결과 #3-CPU shielding effect analysis

CPU shielding /affinity	Data Rate (Hz)	Avg. (us)	Max. (us)	Std. (us)	counts
X	50	64.94	378.01	54.81	30109
	100	64.24	307.92	53.12	30176
O	50	26.93	247.00	26.02	30109
	100	28.65	206.95	26.99	30221

다. 실 환경 네트워크 데이터 저장 실험

마지막으로, 실제 원격운용 환경에서 제안한 시간동기화 방법의 성능을 확인하기 위해 그림 10과 같은 시험환경을 구성하였다. 통제차량과 무인차량 내부는 기가랜 스위치를 통해 유선네트워크로 구성되어있고, 통제차량과 무인차량은 무선통신을 통해 연결하였다. 차량 내 구성품들은 송/수신되는 주요 패킷을 유선네트워크를 통해 기록장치에 전송하고, 기록장치는 이를 수신하여 수신시간과 함께 내부 데이터베이스에 저장한다.

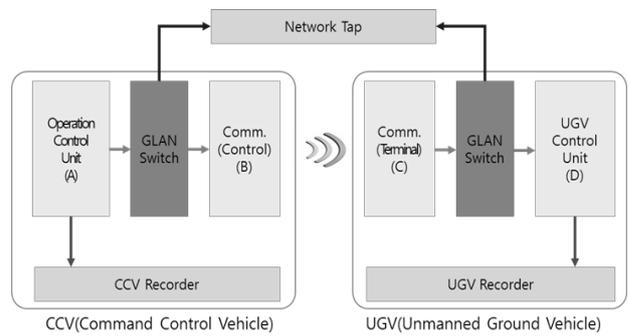


Fig. 10. Test environment set-up for network latency comparison between the Recorder and the network Tap.

그림 10. 기록장치와 네트워크 Tap간의 네트워크 전송지연 비교를 위한 시험 환경구성

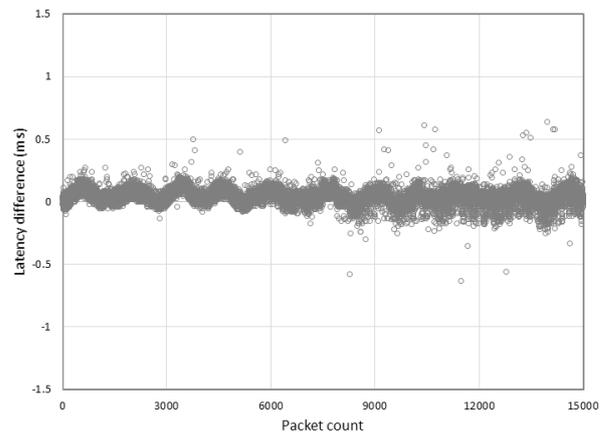


Fig. 11. Network latency difference between the Recorder and the network tap.

그림 11. 기록장치와 네트워크 Tap의 네트워크 전송지연 차이

덧붙여, 기록장치의 패킷 수신시간 기록의 정확도를 비교할 수 있는 대상으로 네트워크 Tap을 이용하였다. 네트워크 Tap은 네트워크 노드 간에 전달되는 트래픽을 미리링하는 외부 모니터링 장치로, 차량 내부 스위치와 연결되어 선택된 포트에 대한 트래픽을 복제하여 기록하

는 동작을 수행한다. 통제차량과 무인차량의 내부 기가 랜 스위치를 하나의 네트워크 Tap장치에 연결하여 패킷을 기록함으로써 별도의 시간동기화 기법없이 동기화된 패킷 수신시간을 기록하였다.

시험을 위해 통제차량에서 무인차량으로 50Hz주기의 신호를 총 15,000개 발생시켰다. 그 후, 통제차량과 무인차량 기록장치로부터 기록데이터를 추출하고, 네트워크 Tap의 기록에서 패킷 수신시간을 추출하여 전송지연을 계산하였다. 그림 11은 두 가지 측정방법의 전송지연 간 차이를 그린 것으로, 1ms 이하의 차이를 보임을 확인할 수 있다. 그림 12에는 각 측정방법의 전송지연 평균과 표준편차를 나타내었다. 기록기를 통한 전송지연은 평균 5.580ms, 네트워크 Tap을 통한 전송지연은 평균 5.623ms를 보였으며, 표준편차는 0.471ms와 0.463ms로 나타났다. 이 결과를 통해 제안하는 시간동기화 기법을 이용한 네트워크 데이터 분석방법은 일반적으로 사용가능한 네트워크 tap 장비와 비교하여도 그 성능 차이가 1ms 이내임을 확인할 수 있었다. 이를 활용하면 네트워크 Tap을 직접 연결하기 힘든 유무선 통신이 복합적으로 적용된 원격운용 시스템의 네트워크 데이터를 효과적으로 분석할 수 있을 것으로 예상된다.

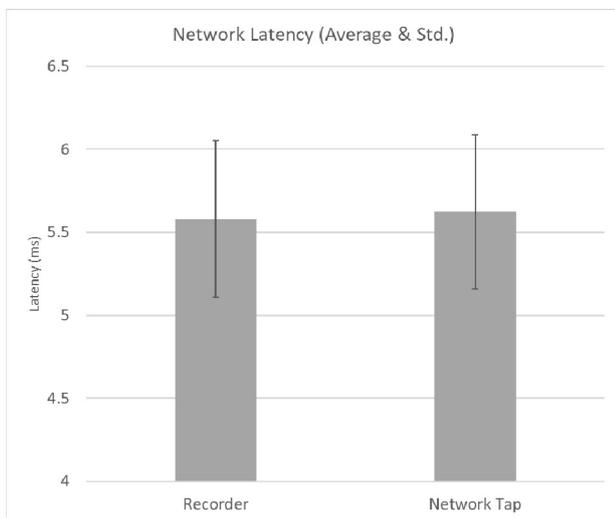


Fig. 12. Network latency (Average & Standard deviation) of the Recorder and the network tap.

그림 12. 기록장치와 네트워크 Tap의 네트워크 전송지연 (평균과 표준편차)

III. 결론

본 논문에서는 통제차량과 무인차량으로 구성된 유무선 복합 네트워크 시스템의 데이터 분석을 위한 시간동

기화 방안을 새롭게 제안하였다. 제안하는 방법은 GPS 신호를 기반으로 수 Km 이상 떨어진 두 개의 시스템에 저장된 네트워크 데이터의 시간을 1ms 이내의 오차로 동기화하였으며, 이를 위하여 시간 tagging 방법 및 네트워크 저 지연 설정 방법 등을 제안하였다. 제안하는 방법의 실효성을 검증하기 위하여 단계적으로 시간 차를 분석하는 실험을 수행하였으며, 제안 방법이 평균 30us 수준의 시간차와 최대 250us 이내의 시간 오차를 가지는 것을 확인하였다. 또한, 통제차량과 무인차량에 장착된 기록장치는 제안하는 시간동기화 방법을 활용하여 네트워크 Tap장비 수준의 데이터 분석이 원활하게 가능함을 확인할 수 있었다. 향후에는 무인차량과 통제차량에서 제안하는 방법을 활용하여 네트워크 데이터를 획득하고 분석하고 개선사항을 찾아서 시간동기화 방안을 개선해 나갈 예정이다.

References

- [1] DongWon Yang, "Detail-Enhanced Target Segmentation Method for Thermal Video Sequences based on Spatiotemporal Parameter Update Techniques," *IET Image Processing*, vol.13, no.1, pp.216-223, 2019. DOI: 10.1049/iet-ipr.2018.5341
- [2] TaeGil Ha, Jongin Lim, DongWon Yang, Jun Hee Lee, Dokyung Kim, Jinyoung Choi, "Moving Object Detection Using Multi-resolution Attention Mechanism," *Journal of the Institute of Electronics and Information Engineers(IEIE)*, vol.56, no.3, pp.81-93, 2019. DOI: 10.5573/ieie.2019.56.3.81
- [3] Jaehoon Choi, Daeyeong Kim, DongWon Yang, Jun Hee Lee, Dokyung Kim, Changick Kim, "Channel Pruning using Scaling Factor of Batch Normalization in Compact Networks," *Journal of the Institute of Electronics and Information Engineers(IEIE)*, vol.56, no.3, pp.292-299, 2019. DOI: 10.5573/ieie.2019.56.3.81
- [4] Dojong Kim, Namgon Kim, DongWon Yang, "Real Time Network Implementation for Military Unmanned Ground Vehicle based on Standard Ethernet," *Journal of KIIT*. Vol.19, No.10, pp. 45-55, 2021. DOI: 10.14801/jkiit.2021.19.10.45
- [5] D. L. Mills, "Internet time synchronization: the network time protocol," *IEEE Transactions on*

Communications, vol.39, no.10, pp.1482-1493, 1991.
DOI: 10.1109/26.103043

[6] M. Langer, K. Teichel, D. Sibold and R. Bermbach, "Time synchronization performance using the network time security protocol," *2018 European Frequency and Time Forum(EFTF)*, pp. 138-144, 2018. DOI: 10.1109/EFTF.2018.8409017

[7] R. L. Scheiterer, C. Na, D. Obradovic and G. Steindl, "Synchronization Performance of the Precision Time Protocol in Industrial Automation Networks," in *IEEE Transactions on Instrumentation and Measurement*, vol.58, no.6, pp.1849-1857, 2009. DOI: 10.1109/TIM.2009.2013655

[8] H. G. Berns and R. J. Wilkes, "GPS time synchronization system for K2K," *1999 IEEE Conference on Real-Time Computer Applications in Nuclear Particle and Plasma Physics*. 11th IEEE NPSS Real Time Conference. Conference Record (Cat. No.99EX295), pp.480-483, Sante Fe, 1999.

[9] H. G. Berns, T. H. Burnett, R. Gran and R. J. Wilkes, "GPS time synchronization in school-network cosmic ray detectors," in *IEEE Transactions on Nuclear Science*, vol.51, no.3, pp.848-853, 2004. DOI: 10.1109/TNS.2004.829368

[10] A. Khalajmehrabadi, N. Gatsis, D. Akopian and A. F. Taha, "Real-Time Rejection and Mitigation of Time Synchronization Attacks on the Global Positioning System," in *IEEE Transactions on Industrial Electronics*, vol.65, no.8, pp.6425-6435, 2018. DOI: 10.1109/tie.2017.2787581

[11] David Baston, "Comparison of Real-time Network Performance of RedHawk™ Linux® 7.5.2 and RedHat® Operating Systems," *A Concurrent Real-Time White Paper*, pp.1-7, 2019.

BIOGRAPHY

DongWon Yang (Member)



2001 : B.S. degree in Electrical Engineering, Seoul National University.

2003 : M.S. degree in Electrical Engineering, Seoul National University.

2016 : Ph.D. degree in Electrical Engineering, KAIST.

2003~Present : Principal Researcher, Agency for Defense Development(ADD).

Namgon Kim (Member)



2004 : B.S degree in Computer Engineering, Chonnam National University.

2006 : M.S degree in Info. and Comm. Engineering, GIST.

2012 : Ph.D. degree in Info. and Mech. Engineering and GIST.

2019~Present : Senior Research Engineer, Agency for Defense Development(ADD).

Dojong Kim (Member)



1987 : MS degree in Electrical Engineering, Kyungpook National University.

2001: Ph.D. degree in Electrical Engineering, KAIST.

1987~Present : Principal Researcher, Agency for Defense Development (ADD).