

무인 수상정 전자 장치를 위한 통신 미들웨어 설계 및 구현

(Design and Implementation of a Communication Middleware for Electronic Devices of Unmanned Surface Vehicle)

배종윤*, 최훈**

(JongYoon Bae, Hoon Choi)

요약

본 논문에서는 카메라 및 다양한 센서가 복합되어 고속의 데이터를 다중으로 처리하는 성능을 요구하는 전자광학장비의 안정적인 데이터 전송을 위해 Event 기반의 동기화 방식을 통한 Multi-Thread 환경의 다중 통신 미들웨어 설계 및 구현 방법을 제안하였다. 구현된 통신 미들웨어의 성능 검증을 위해 영상 데이터 및 센서 데이터를 전송하여 송신 주기 대비 수신 주기에 대한 차이를 비교하고, 다중으로 전송 및 처리할 수 있는 최대 통신 가능 수를 측정 및 분석하였다. 또한 전송되는 데이터의 무결성 검증과 Round Trip Time 측정 등의 실험을 통해 제안하는 통신 미들웨어의 성능을 검증하였다.

■ 중심어 : 통신 미들웨어 ; 전자 광학 장비 ; 멀티 스레드

Abstract

In this paper, designing and implementing multi-communication middleware in multi-thread environmet through event-based synchronization method are proposed for stable data transmission of electronic optical equipment, which requires combining camera and various sensors to process multiple high-speed data. To verify the performance of the implemented communication middleware, image data and sensor data were sent to compare differences in reception-based and transmission-based cycles, and the maximum number of communication possibilities to transmit and process multiple was measured and analyzed. In addition, the proposed communication middleware's performance was verified through experiments such as validating the integrity of the transmitted data and measuring the Round Trip Time.

■ keywords : Communication Middleware ; Electronic Optical Equipment ; Multi-Thread

1. 서론

전자광학장비는 영상 신호를 획득하고 사용자에게 전달하여 목표물에 대한 정보를 제공하는 장비이다. 전자광학장비의 모태가 되는 광학장비는 원거리의 목표물을 확인하는 용도로 사람이 직접 착용하거나 조작하는 형태의 망원경이 그 시초라 할 수 있다[1]. 최근에는 센서 기술의 발달로 영상 획득 센서 외에도 거리 측정 센서를 통한 목표물과의 거리 측정과 Pan/Tilt 모터의 장착으로 상하좌우로 회전하며 각도로 영상을 확인하는 등, 사용 목적에 따라 더욱 다양한 센서가 적용될 수 있어 기술의 활용 범위 또한 넓다고 할 수 있다.

현재는 정보통신 기술의 발달로 인해 유/무선 네트워크를 통해 원격지에서 전자광학장비를 제어하거나 정보를 획득하고 영상을 감시 가능하여 무인기나 무인선 등에 탑재되는 추세이다. 다양한 전자광학장비의 종류 중에서도 주로 무인 수상정에 탑재되는 전자광학추적장비(EOTS)는 광학 센서를 통해 목표물을 추적하면서 안정화된 영상을 관측자에게 전달하고 표적 정보를 전달하여 사격 통제 시스템과의 연동을 제공하는 등의 기능을 수행한다[2]. 이처럼 여러 센서 그리고 타 체계와의 다중 통신을 관리하며 산발적으로 발생하는 데이터들을 안정적이고 신뢰성 있게 처리하는데, 통신 미들웨어는 매우 중요한 역할을 한다. 기존의 연구들은 영상 추적 알고리즘과 안정화 제어기법 [3, 4, 5] 등에 초점이 맞추어져 있어 이러한 다중 통신을 담당하는 인터페이스 성능 검증에 대한 연구는 진행되지 않았다.

* 학생회원, 충남대학교 컴퓨터공학과

** 정회원, 충남대학교 컴퓨터공학과 교수

※ 본 연구의 최훈은 한국전력공사의 2018년 착수 에너지 거점대학 클러스터 사업에 의해 지원받았음 (과제번호:R18XA05)

접수일자 : 2019년 05월 30일

게재확정일 : 2019년 09월 18일

수정일자 : 1차 2019년 08월 14일

교신저자 : 최 훈, e-mail : hc@cnu.ac.kr

본 논문에서는 다중 통신을 위한 통신 미들웨어의 설계 및 구현 방법에 대해 설명하고 실험을 통해 성능을 검증하였다.

본 논문의 구성은 2장에서는 관련 연구로서 전자광학장비와 통신 미들웨어 그리고 Multi-Thread에 대해 알아보고, 제안하는 통신 미들웨어의 설계 및 구현 방법에 대해 기술한다. 3장에서는 실험을 통해 제안하는 통신 미들웨어의 성능을 검증하며, 끝으로 4장에서는 결론 및 향후 연구 방향에 대해 기술한다.

II. 설계 및 구현

1. 관련연구

가. 전자광학장비

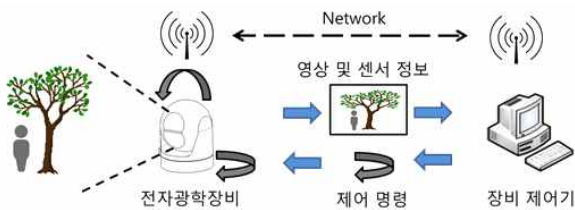


그림 1. 전자광학장비 시스템 구성도

전자광학장비는 주로 군사적인 목적으로 개발되어 왔으며 국내에 최초로 개발된 전자광학장비는 1983년에 보급된 양안형 야간투시경으로 저조도의 상황에서 물체에 반사되는 미약한 빛을 증폭시키는 미광 증폭식 장비였다[1]. 이처럼 사람의 눈으로는 식별할 수 없는 상황을 센서 기술을 통해 빛 에너지의 증폭 또는 목표물이 지니고 있는 열에너지와 주변의 온도 차이를 감지하는 센서, 또는 레이저를 투사하여 무월광(無月光)의 상태에서도 대낮과 같은 식별이 가능하도록 영상을 제공하는 등의 첨단 기술이 접목되어 개발되고 있다. 현대의 전쟁 양상은 정보전으로 변화하고 있어 전장의 상황을 상대방보다 먼저 확인할 수 있는 첨단 기술력의 확보가 중요시되며 이러한 작전 상황에 따라 전자광학장비도 초창기의 모습과는 많은 차이를 보인다. 근래에는 영상 획득 센서와 함께 Pan-Tilt Unit을 포함하고 특정 기구에 통합하여 원격 제어를 지원하는 감시 장비 형태로 개발되고 있으며, 탐지 목적에 따라 수십 km 거리의 목표물을 확인하기 위해 전자광학장비에 전동 줌 렌즈가 접목되기도 하며, 과장 대역이 다른 영상 획득 센서를 복합하는 등 사용 목적에 따라 다양한 영상 감시 임무를 수행한다.

나. 통신 미들웨어

통신 미들웨어는 분산 컴퓨팅 환경에서 광범위하게 분산된 객체 간에 원만한 통신을 제공하기 위해 사용되며, 서로 다른

운영체제와 이종의 프로토콜을 사용하는 네트워크 간의 호환성을 목적으로 분산 시스템을 구성하는데 필수적인 요소이다. 통신 미들웨어의 장점에는 표준화된 인터페이스의 제공이 가능하고 다중 통신 시 부하의 분산을 위한 스케줄링이 가능한 점, 분산 업무를 동시에 처리가 가능하다는 점 등이 있다.

다. Multi-Thread

Thread란 1개의 Process 내에서 실행되는 흐름의 단위를 말하며 Process는 1개의 프로그램이 사용하는 작업 단위이다. Process 하나는 기본적으로 1개의 Thread를 갖지만 1개 이상의 Thread로 Process가 구성되는 경우 이를 Multi-Thread라고 부른다. Multi-Process와 Multi-Thread는 여러 개의 작업을 병렬로 처리한다는 점에서 동일한 기능을 수행한다고 할 수 있으나 메모리 관리의 측면에서 보면 Multi-Thread가 효율적이다. Process가 포함하는 Thread의 수가 증가할 때 Process의 수가 증가할 때보다 적은 메모리가 필요하기 때문이다. 이처럼 Multi-Thread는 Multi-Process보다 효율적인 자원 사용이 가능하고, Thread 간의 전환 속도가 Process 간의 전환보다 빠르다는 장점이 있다[6].

(1) 동기화 문제

Multi-Thread 환경에서는 각각의 Thread가 공유 자원에 접근해야 하는 경우가 발생하는데 2개 이상의 Thread가 동시에 공유 자원에 접근하는 경우에 동기화 처리를 하지 않으면 경쟁 상태나 교착 상태(DeadLock)[7]와 같은 동기화 문제가 발생할 수 있다. 이와 같은 상황이 되면, 의도하지 않은 결과를 발생시키거나, 이미 하나의 Thread가 자원을 점유한 상태로 자원을 해제하지 않아 다른 Thread가 무한정 대기하는 상태가 되는 등 공유 자원의 정상적인 사용이 불가능하게 된다[8]. 따라서 이러한 문제를 해결하기 위한 Thread 간의 동기화는 매우 중요하다.

(2) 동기화 기법

Multi-Thread 동기화 기법은 유저 모드 동기화와 커널 모드 동기화로 나눌 수 있는데 유저 모드 동기화는 응용 프로그램 단에서 소프트웨어(SW)적으로 현재 수행 중인 Thread의 상태를 보호하여 다른 Thread의 접근을 막아주는 형식이다. 대표적인 방법에는 임계 영역(Critical Section)과 InterLock 등이 있다. 커널 모드 동기화는 동기화의 관리를 커널이 대신하여 개발자의 잘못된 구현 방식으로 야기할 수 있는 교착상태를 피할 수 있다는 장점이 있으며, 상호배제(Mutex), Semaphore, Event 기반 동기화 등의 동기화 기법이 있다.

2. 통신 미들웨어(CMLib) 설계

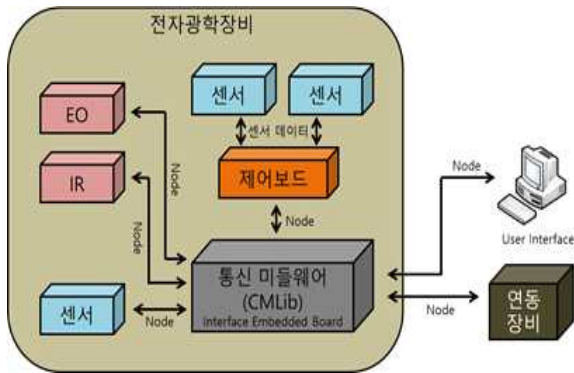


그림 2. 전자광학장비 시스템 구성 예

그림 2는 통신 미들웨어(CMLib)가 전자광학장비 내부의 인터페이스를 담당하는 특정 임베디드 보드에 설치되어 전자광학 장비 내부의 각종 센서 데이터 및 영상 데이터를 처리하고 외부의 연동 장비 및 User Interface SW와의 연동을 나타내는 그림이다. 그림과 같이 통신 미들웨어(CMLib)는 다수의 센서 및 연동 장비와 통신을 하면서 발생하는 각각의 데이터를 처리하는 역할을 하는데 이처럼 고속의 영상 데이터와 다중 센서 데이터의 처리를 Single-Thread로 처리하게 된다면 순차적으로 데이터를 처리하게 되므로 지연이 발생하게 되고 통신 채널 수와 데이터의 크기가 증가함에 따라 처리 성능은 저하하게 된다. 따라서 다중의 데이터 처리 성능을 향상하기 위해서는 데이터의 병렬 처리가 가능한 Multi-Thread 환경 구성이 불가피하다.

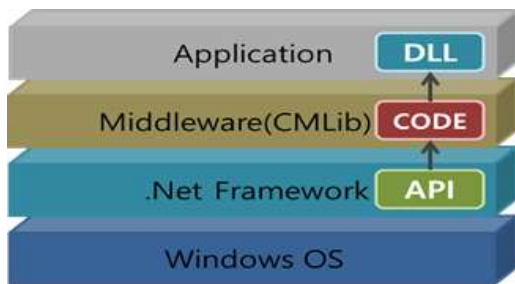


그림 3. CMLib을 활용한 Application 개발 구성도

CMLib은 Communication Middleware Library의 약어이며, 그림 3과 같이 .Net Framework API를 참조하여 통신 관리 및 데이터 처리 기능을 제공하는 통신 미들웨어로 윈도우즈 (Windows) 운영체제 상에서 구현되었다. CMLib는 다중 통신과 Multi-Thread 환경의 데이터 처리 기능에 특화되어있으며 Application 개발자에게 통신에 대한 사항을 고려하지 않고 개발을 가능하게 함으로써 개발 시간의 단축과 적은 비용으로 효율적인 Application을 개발 가능한 장점이 있다. 또한 CMLib

은 DLL(Dynamic Linking Library)의 형태로 구현되어 있어 재사용과 Application 개발의 용이성을 고려하였다.

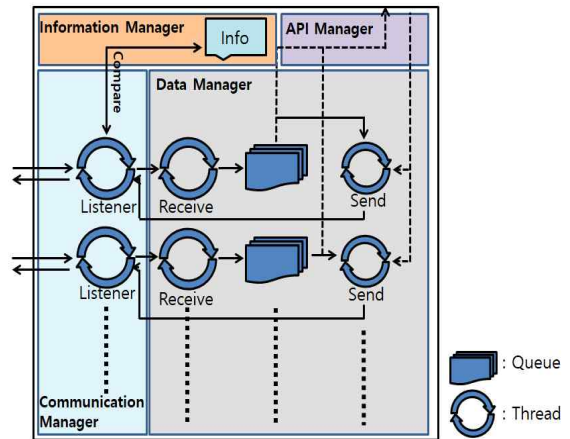


그림 4. 통신 미들웨어(CMLib) 내부 구조

그림 4는 CMLib의 내부 구조이며, 각 통신 채널의 데이터 처리를 위해 통신 서버 소켓 1개와 2개의 클라이언트 소켓, 1개의 Data Queue 그리고 접근하는 클라이언트의 확인을 위한 서버 Listener Thread, 수신 데이터 확인을 위한 Data Receive Thread, 데이터의 송신을 위한 Data Send Thread로 구성하였다. 앞서 설명한 Multi-Thread 동기화 문제는 공유 자원에 여러 Thread의 접근이 필요한 경우 발생하며, 접근 Thread의 수가 증가함에 따라 동기화 처리를 위한 복잡도도 증가하기 때문에 CMLib에서는 1개의 통신 채널에 각각 1개의 송/수신 Thread를 할당하여 동기화 문제를 최소화하였다.

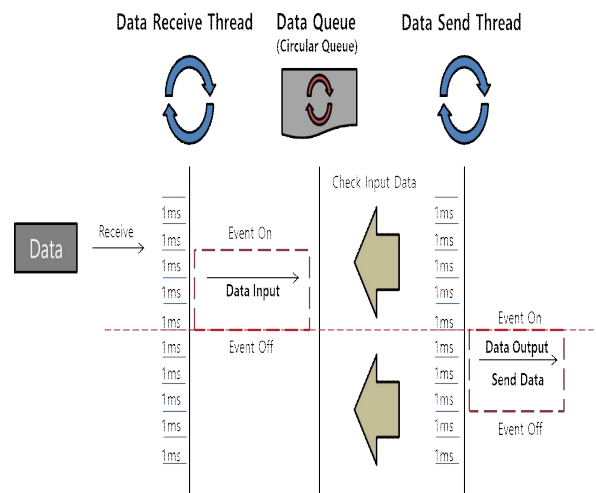


그림 5. 각 채널 별 Thread의 동작 구조

그림 5는 CMLib의 각 1개의 통신 채널에서 동작을 수행하는 Thread에 대한 그림으로 앞서 설명한 Multi-Thread의 동기화 기법 중, Event 기반의 동기화를 적용하였다. 공유 자원인 Queue는 하드웨어 성능이 제한적인 임베디드 보드의 효율적인 메모리 사용을 위해 Circular Queue로 구성하였다. 각 Thread에서는 데이터의 입/출력을 위해 Circular Queue의 가용 상태를 체크하고 데이터 수신 시, 대기 중이던 Event 객체가 활성화되고 데이터의 입력을 마치는 즉시, 출력을 위한 Event를 호출하여, 다음 Thread에서의 동작을 수행할 수 있게 함으로써, 각 Thread 간의 동작 소요 시간을 최소화하였다. 그림 5와 같은 Thread의 동작 구조를 통해 각 채널에서는 Queue의 데이터 입력과 출력이 동시에 일어나지 않도록 하여 동기화 문제를 해결하였다. 또한 각 Thread에는 최소한의 여유 시간으로 1ms를 설정하여 하나의 Thread에서 CPU 자원을 계속하여 독점하는 문제를 피하였고, Queue의 가용 상태를 빠르게 확인을 할 수 있도록 하였다.

3. 통신 미들웨어(CMLib) 구현

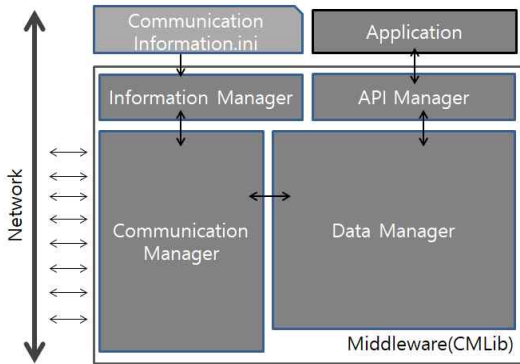


그림 6. 통신 미들웨어(CMLib) 구성

그림 6은 제안하는 통신 미들웨어(CMLib)를 구성하는 4개의 관리자에 대한 것으로 API Manager를 통해 응용 프로그램 개발자는 CMLib와 인터페이스 하여 데이터에 접근이 가능하다. Information Manager는 외부의 통신 정보 파일에 접근하여 기록된 정보를 통해 Server-Client 및 통신에 필요한 Thread 등의 자원을 할당하는 역할을 한다. Communication Manager는 접근하려는 Client의 정보를 통신 정보 파일에 기록된 정보와 비교하여 승인된 Client와 세션을 생성하고 Data Manager가 데이터를 처리할 수 있도록 송/수신 Thread를 활성화한다. Data Manager는 통신 채널 별로 각각의 Receive/Send Thread와 Queue를 구현하여 접속한 Client의 전달되는 데이터를 처리하는 역할을 한다.

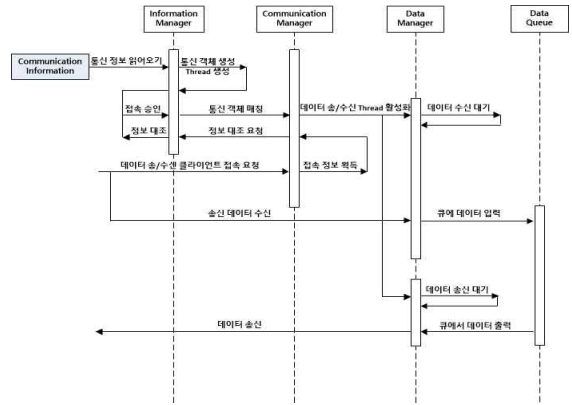


그림 7. 통신 미들웨어 시퀀스 다이어그램

그림 7은 CMLib의 동작을 설명하는 전체 시퀀스 다이어그램으로 Application을 통한 CMLib의 초기화 시, 외부의 특정 위치에 기록된 통신 정보 파일을 읽어 들여 현재 프로젝트에 사용하게 될 다중 통신의 수와 통신 타입, IP Address, Port 등의 정보를 참조하여 통신 객체 및 Thread 객체를 동적으로 생성한다. 생성된 Thread 중, 통신 접속 모니터링을 위한 Listener Thread는 자동으로 활성화되고 이후, 클라이언트의 접속 요청이 발생하면 Listener Thread에서 클라이언트의 통신 정보를 획득하고 CMLib 초기화 시, 읽어 들였던 통신 정보 파일의 내용과 정보를 대조한다. 비교 결과 통신 정보가 일치하면 승인된 클라이언트로 판단하여 미리 정의해두었던 통신 객체에 클라이언트를 매칭시키고 세션을 맺는다. 이후, 데이터의 송/수신을 위한 Data Send Thread, Data Receive Thread가 활성화되어 데이터의 수신을 대기한다. 데이터 수신이 발생하면 Data Receive Thread에서 읽어 들이고 Queue에 데이터를 입력한다. 이때, 주기적으로 Queue를 확인하고 있던 Data Send Thread에서는 데이터의 입력이 발생하면 Queue에서 데이터를 출력한 뒤, 데이터를 송신하는 순서로 진행된다.

III. 실험 및 성능 검증

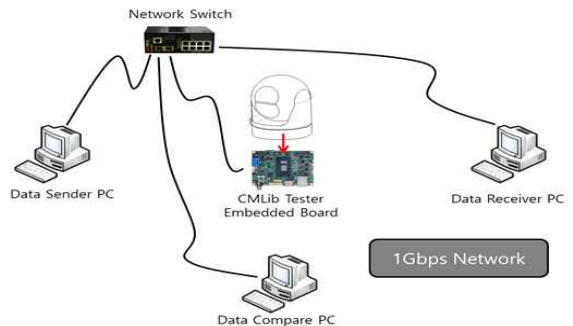


그림 8. 실험 구성도(하드웨어)

본 논문에서 제안하는 CMLib의 성능 검증을 위해 Data Sender Application, CMLib Test Application, Data Receiver Application, Data Compare Application을 추가 구현하였으며, 각 Application의 구동을 위한 실험 하드웨어를 그림 8과 같이 준비하였다. 실험을 위해 준비한 데이터는 표 2와 같으며, 센서 데이터를 모사한 문자 데이터는 보통 1 패킷 기준 200 bytes 이하의 데이터를 전송하지만, 상황에 따라 더 큰 데이터를 전송할 수 있어야 하며, 영상 데이터의 경우는 압축 기술과 영상 품질에 따라 전송 데이터가 달라질 수 있으나, 보편적으로 많이 사용되는 H.264 코덱 기준으로 적게는 512Kbps에서 4Mbps의 Bitrate 을 사용하기 때문에 표 1의 데이터는 실제 전자광학장비에서 사용되었던 인터페이스 임베디드 보드의 성능을 고려하여 선정하였다. 데이터의 전송 주기는 NTSC 표준인 30fps(frame per second)를 기준으로 하였으며, 해상도 사이즈 별로 30fps로 저장된 1분짜리 동영상을 준비하여, Data Sender Application에서 매 1 frame의 영상 데이터(60초 x 30 frame = 1,800 frame)를 캡처 및 전송하고, CMLib을 통해 Data Receiver Application에서 수신하는 절차로 실험하였다.

$$1 \text{ frame} = \text{해상도} * \text{픽셀 채널 수(RGB=3)} * \{\text{Bit}(8)\}$$

$$\text{초당 전송 데이터 크기} = 1 \text{ frame} * \text{주기}(30\text{Hz})$$

표 1. 전송되는 데이터 크기

타입	데이터 크기		
	1패킷 기준	1초 기준 (Bytes)	1초 기준 (Bits)
문자	1,000 Bytes	30,000 Byte/s	240,000 Bit/s
영상 (100x 100)	30,000 Bytes	900,000 Byte/s	7,200,000 Bit/s
영상 (320x 240)	230,400 Bytes	6,912,000 Byte/s	55,296,000 Bit/s
영상 (640x 480)	921,600 Bytes	27,648,000 Byte/s	221,184,000 Bit/s

1. 실험 환경

본 실험의 하드웨어는 1Gbps의 네트워크를 지원하는 네트워크 스위치와 PC로 구성하였으며, 표 2와 같은 하드웨어 환경에서 통신 미들웨어(CMLib)의 성능 검증을 진행하였다.

표 2. 실험 하드웨어 스펙

Hardware	OS	CPU	RAM
Data Sender PC	Windows 7 (64bit)	i5-4210U (1.70GHz)	4GB
CMLib Test Embedded Board	Windows 8 (64bit)	i7-6600U (2.6GHz)	4GB
Data Receiver PC	Windows Server 2012 R2 (64bit)	E5-2699 (2.3GHz)	32GB
Data Compare PC	Windows 10 (64bit)	i7-7740X	8GB

그림 9에서 (A)는 문자 및 영상 데이터를 생성하고 30Hz로 전송하는 Data Sender Application, (B)는 CMLib 기반으로 구현된 CMLib Test Application, (C)는 Data Receiver Application, (D)는 Wireshark 프로그램으로 Data Sender Application 측의 송신 주기 측정과 CMLib Test Application을 거쳐 Data Receiver Application 측이 수신한 데이터들의 수신 주기를 모니터링하기 위해 사용되었다.



그림 9. 성능 검증 소프트웨어

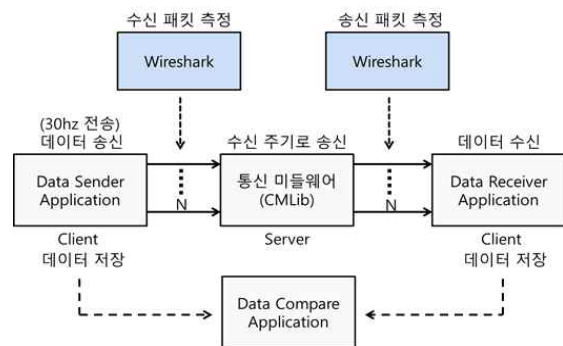


그림 10. 실험 구성도(소프트웨어)

그림 10은 소프트웨어 실험 구성도이다. 각 Application은 최대 30채널의 다중 통신이 가능하도록 구현하였으며, 데이터의 크기와 채널 수를 변경하여 구현된 실험 Application을 통해 각 실험에서 제시하는 조건을 만족하는 최대 다중 통신 가능 채널 수를 확인하는 것을 목표로 하였다. 실험의 주제는 3가지로 구성하였으며, 각 실험은 다시 2가지 실험으로 분류하여, 실험 1은 실제 전자광학장비의 상황을 모사한 영상 2채널과 센서 8채널 통신 실험, 실험 2는 CMLib이 처리할 수 있는 최대 성능을 확인하기 위한 실험으로 구성하였다.

첫 번째 실험으로 CMLib의 데이터 처리 성능 분석 실험은 통신 채널 수와 데이터의 크기가 증가함에 CMLib에서 처리되는 데이터의 처리 지연을 검증하기 위한 실험으로 송신 주기 대비 수신 주기의 비교와 수신되는 각 1,800개 패킷의 간격을 분석하여 안정적인 데이터 수신이 가능한 다중 채널 수를 확인한다. 두 번째로 Round Trip Time 분석 실험은 1,800개의 각 패킷의 Round Trip Time이 송신 주기인 30Hz(33ms) 이내에 형성되는 것을 안정적인 조건으로 제시하여 이를 만족하는 다중 통신 채널 수를 확인한다. 세 번째로 무결성 검사를 통해 Event 기반의 Multi-Thread로 구현된 CMLib의 동기화 문제 혹은 설계상의 오류로 인한 데이터의 오염 및 누락이 발생하지 않고 신뢰성 있는 데이터의 송/수신이 이루어졌는지 확인한다.

2. 데이터 처리 성능 분석 실험

가. 실험

전자광학장비는 사용 목적에 따라 포함되는 센서와 외부와의 연동 등 통신 노드의 수가 달라질 수 있다. 본 실험은 두 개의 실험으로 구성하였으며, 첫째는 기본적으로 전자광학장비에 구성되는 카메라 2채널과 센서 및 외부 연동 8채널의 사용 예를 모사하여 100x100(7.2Mbps) 해상도의 영상 데이터 2채널과 1패킷당 1,000 Byte 크기의 문자 데이터 8채널을 전송하여 송신 주기와 수신 주기의 차이점이 발생하는지 확인한다. 두 번째 실험은 전송 데이터의 크기와 채널 수의 증가에 따라 CMLib의 각 채널에서 동작하는 Thread들의 데이터 처리 시간이 증가하기 때문에, 각 채널에서 처리되는 패킷들의 일률적인 처리 성능을 제시하기 위해 표 3과 같이 송신 주기인 30Hz 즉, 송신 간격(33ms)을 기준으로 2%, 5% 10% 범위의 조건을 제시하여 CMLib을 통해 전달되는 1800 패킷 사이의 수신 간격이 각 허용 범위 조건을 초과하는 횟수가 1%(18개) 이하로 측정되는 것을 기준으로 데이터 크기 별 최대 통신 가능 수를 확인한다.

표 3. 데이터 처리 지연 허용 범위 조건

기준	2%	5%	10%
33ms	33.66ms이내	34.65ms이내	36.3ms이내

나. 결과

표 4는 송/수신 주기 비교를 위한 첫 번째 실험의 결과로 표의 결과 값을 통해 기본적인 구성의 전자광학장비에서 다중 데이터 전송 시, 송/수신 주기 차이가 거의 나타나지 않음을 확인할 수 있으며, 이를 토대로 실제 전자광학장비에서의 데이터 전송 시에도 동일한 네트워크 상황이라면 실험 결과와 같은 성능의 안정적인 데이터 전달이 가능할 것으로 판단할 수 있다.

표 4. 혼합 데이터 10채널 데이터 처리 성능 결과

채널	송신주기	수신주기
채널1(영상)	30.32Hz	30.32Hz
채널2(영상)	30.32Hz	30.32Hz
채널3(문자)	30.27Hz	30.27Hz
채널4(문자)	30.28Hz	30.28Hz
채널5(문자)	30.28Hz	30.28Hz
채널6(문자)	30.28Hz	30.28Hz
채널7(문자)	30.28Hz	30.28Hz
채널8(문자)	30.28Hz	30.28Hz
채널9(문자)	30.28Hz	30.28Hz
채널10(문자)	30.28Hz	30.28Hz

그림 11과 그림 12는 CMLib의 일률적인 데이터 처리 성능을 제시하기 위한 두 번째 실험의 결과로 각각 100x100 해상도와 320x240 해상도의 영상 데이터를 다중으로 전송하였을 때, 채널별로 각 패킷의 수신 간격이 2%, 5%, 10%의 조건들을 초과하는 횟수가 1%(18개)를 넘어서는 경우의 다중 통신 채널 수와 조건 초과 채널 수를 나타낸 그래프이다.

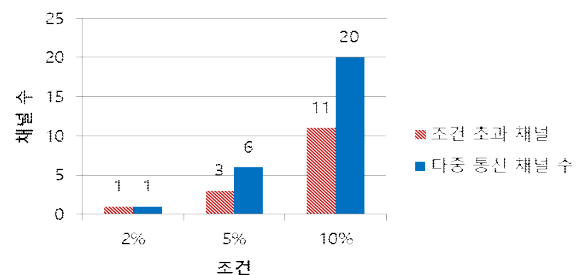


그림 11. 100x100(7.2Mbps) 다중 통신 조건 초과 그래프

그림 11의 그래프를 통해 100x100(7.2Mbps) 해상도의 영상 데이터 다중 통신 시, 각 패킷의 수신 간격 중, 5%(34.65ms)의 조건을 초과하는 횟수가 1%(18개) 이상인 다중 채널 수는 6개 중 3개, 10%(36.3ms)의 조건을 초과하는 채널 수는 20개 중 11개로 나타났으며, 각 조건에서 1채널씩을 감소하여 5%는 5채널, 10% 19채널로 실험을 한 결과, 각 조건을 초과하는 횟수가 18개를 초과하는 패킷은 발생하지 않았다. 이 결과를 통해 CMLib은 다중 5채널 통신 상황에서 34.65ms 이내의 간격 그리고 다중 19채널 통신 상황에서 36.3ms 이내의 간격으로 일괄적인 데이터의 처리가 가능함을 확인하였다.

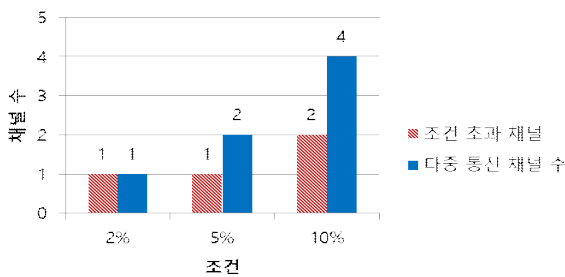


그림 12. 320x240(55Mbps) 다중 통신 조건 초과 그래프

그림 12의 그래프를 통해 320x240(55Mbps) 해상도의 영상 데이터는 다중 채널별 패킷의 수신 간격 중, 5%(34.65ms)를 초과하는 다중 채널 수는 2개 중 1개, 10%(36.3ms)를 초과하는 다중 채널 수는 4개 중 2개로 나타났으며, 앞서 실험한 100x100(7.2Mbps) 해상도의 데이터보다 처리해야 하는 데이터 크기가 증가하여 다중 통신이 가능 채널 수가 감소함을 알 수 있다. 마찬가지로 각 조건에서 1채널씩 감소하여 5%는 1채널, 10%는 3채널로 실험을 한 결과, 1%(18개)를 넘어서는 패킷은 발생하지 않았다. 이 결과를 통해 CMLib은 다중 3채널 통신 상황에서 320x240(55Mbps) 사이즈의 데이터를 36.3ms 이내의 간격으로 처리 가능함을 확인하였다.

그림 11과 그림 12의 그래프에서 2%의 조건은 1개의 채널에서도 조건을 초과하는 결과를 볼 수 있는데 Thread 내의 반복문이 CPU 자원을 독점하는 것을 막기 위해 1ms의 지연 시간을 설정해준 것의 이유로 33ms의 2%는 33.66ms이고 33ms에 1ms의 지연 시간이 더해지는 경우 2%의 조건을 초과하기 때문에 발생하는 결과였다. 또한 640x480(221Mbps) 해상도의 영상 데이터는 채널 1개의 전송만으로 이미 표 4에서 제시한 조건의 10%를 초과하여 다중 통신 가능 수를 확인하는 것이 무의미하여 결과에서 제외하였다.

이 실험을 통해 데이터 크기의 변화에 따라 다중 통신 가능 채널의 수가 제한되는 것을 확인할 수 있는데, 이는 채널 수와 데이터 크기가 증가함에 따라 CMLib의 채널별 Thread가 각각의 패킷을 수신하고 데이터를 Queue에 입/출력 및 전송하는

데까지 걸리는 처리 시간이 동시에 증가하기 때문에 실험과 같이 조건을 제한하여 일괄적이고 안정적인 데이터 처리를 위한 데이터 크기별 통신 가능 채널 수를 제시하였다.

3. RTT(Round Trip Time) 분석 실험

가. 실험

현장에 설치되어 사용되는 전자광학장비의 경우 네트워크 환경과 품질에 따라 그 성능의 차이가 발생할 수 있다. 본 실험에서는 1Gbps의 안정적인 유선 네트워크 상황에서 다중 통신 시, 측정되는 CMLib의 RTT를 확인하여 성능을 제시하였다. 2절의 실험과 마찬가지로 첫 번째 실험은 전자광학장비의 구성을 모사하여 100x100(7.2Mbps) 해상도 크기의 영상 데이터와 1000Bytes의 센서 데이터를 다중 10채널 통신을 통해 채널별로 송신되는 1,800개 패킷의 RTT를 측정하고 송신 주기인 30Hz(33ms) 이내에 형성되는지 확인하였다. 두 번째 실험은 데이터 크기 별로 다중 통신 상황의 모든 통신 채널의 RTT가 33ms 이내로 형성되는 최대 다중 통신 채널 수를 확인하였다.

나. 결과

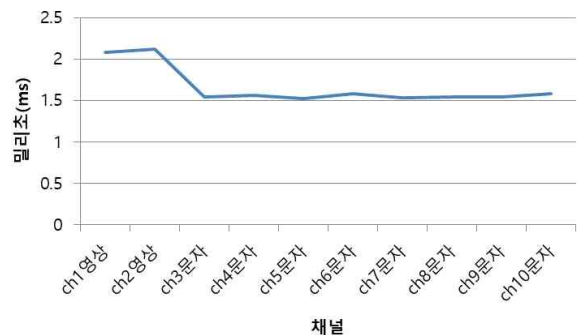


그림 13. 채널 별 통신 데이터의 평균 Round Trip Time

그림 13은 첫 번째 실험의 결과로 다중 10채널 혼합 데이터의 채널 별 송신되는 1,800개의 패킷에 대한 RTT의 평균을 그래프로 나타낸 것이다. 앞선 데이터 처리 성능 실험의 결과를 통해 CMLib의 채널별 Thread는 처리되는 데이터 크기에 따라 처리 시간이 소요됨을 확인하였고, 그 영향으로 문자 데이터에 비해 크기가 큰 영상 데이터를 처리한 채널 1과 채널 2의 RTT가 상대적으로 크게 형성된 것을 알 수 있다. CMLib을 통해 전송되는 모든 패킷은 송신 주기인 30Hz(33ms) 이내로 처리가 되면 사용자 입장에서는 영상 및 센서 데이터를 확인하는 것에 부자연스러움을 느낄 수 없다. 채널별 패킷 중, RTT가 최대로

측정된 것은 영상의 경우 3.3ms였고, 문자의 경우 3.076ms로 송신 주기 대비 9배 이상 빠른 속도로 데이터의 처리가 가능한 것을 확인할 수 있다. 또한 그림 13의 평균 RTT 그래프를 통해 데이터별로 영상의 경우 평균 2.3ms 이내, 문자의 경우 평균 1.7ms 이내의 안정적인 RTT이 형성되는 것을 확인하였다. 이 실험의 결과를 통해 동일 네트워크 환경이라면 CMLib을 통해 영상 및 센서 데이터를 송신 주기 이내에 목적지로 전송이 가능하다는 것을 알 수 있다.

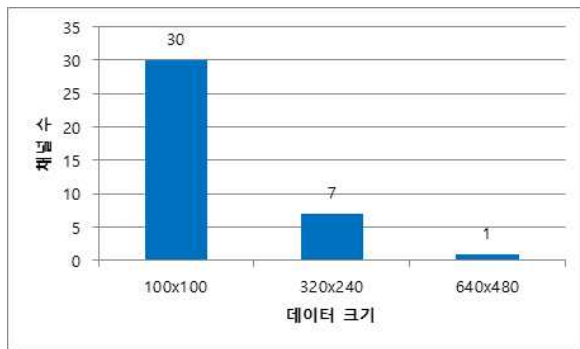


그림 14. RTT 33ms 이하의 다중 통신 가능 채널 수

그림 14는 두 번째 실험의 결과로 다중 채널 통신 상황에서 채널 별로 1,800개의 모든 패킷이 33ms 이하의 RTT를 만족하는 데이터 크기 별 최대 다중 통신 채널 수를 나타낸 그래프이다. 분석 결과, 100x100(7.2Mbps) 사이즈의 영상 데이터는 구현된 최대 통신 가능 수인 30채널의 다중 통신에서도 모든 패킷이 최대 10ms 이하의 시간으로 RTT이 나타나는 것을 확인하였다. 320x240(55Mbps) 사이즈에서는 7개 채널까지 최대 15ms 이하의 시간으로 모든 패킷의 RTT이 형성되었고, 8개 채널에서는 채널의 양상에 따라 다른 추이를 보이며 33ms의 시간을 초과하였다. 640x480(221Mbps) 해상도에서는 채널 1개의 통신에서 최대 20ms 이하의 RTT이 형성되었고, 채널 2개에서부터는 기본적으로 형성되는 RTT이 20ms 이상으로 나타나며 33ms의 조건을 초과하였다. 전자광학장비는 고속의 데이터 전송과 처리가 필요하며, 영상의 관측이 주목적이기 때문에 끊임 없는 영상의 확인을 위한 조건으로 RTT의 기준을 33ms 이내로 설정하였고, 실험 결과를 통해 100x100 사이즈에서 30 채널, 320x240에서는 7채널, 640x480 사이즈는 1채널로 조건을 만족하는 다중 통신이 가능함을 확인하였다.

4. 데이터 무결성 검사

Data Compare Application을 통해 각 PC에 저장된 폴더를 읽어서 문자 데이터의 경우 텍스트 파일로, 영상 데이터의 경우

이미지 파일을 참조하여 SSD(Sum of Squared Difference) 알고리즘에 의해 계산된 결과 값을 확인하여 무결성을 검증한다. SSD는 주로 두 이미지의 Pixel을 비교하는 영상 정합 기법으로 두 이미지의 처음부터 끝까지 서로 같은 픽셀 좌표의 차를 구해서 제공한 값의 모든 합을 통해 이미지의 유사성을 측정하며, 완전히 동일한 경우 합의 값은 0으로 나타나게 된다.

$$\sum_{xy} (I_1(x,y) - I_2(x,y))^2 \tag{1}$$

가. 실험

2절의 실험과 마찬가지로 첫 번째 실험은 전자광학장비의 구성을 모사한 10채널(영상 2채널 + 문자 8채널) 대한 무결성 검사를 진행하였고, 두 번째 실험은 앞서 데이터 처리 성능 분석 실험의 조건을 만족한 데이터 크기 별 통신 채널 수를 기준으로 실험을 진행하였다.

나. 결과

표 5. 10채널 통신 데이터 무결성 검사 측정 결과

채널	무결성 결과 값
채널1(영상)	0바이트
채널2(영상)	0바이트
채널3(문자)	0바이트
채널4(문자)	0바이트
채널5(문자)	0바이트
채널6(문자)	0바이트
채널7(문자)	0바이트
채널8(문자)	0바이트
채널9(문자)	0바이트
채널10(문자)	0바이트

표 6. 처리 성능 조건 만족 채널 무결성 검사 측정 결과

이미지 크기	5% 조건 만족 채널	10% 조건 만족 채널	무결성 결과 값
100x100	5채널	19채널	0바이트
320x240	1채널	3채널	0바이트
640x480	0채널	1채널	0바이트

표 5와 표 6은 각 실험에 대한 무결성 실험의 결과로 SSD 알고리즘을 통한 두 파일의 바이트 비교 합이 0바이트로 측정되어 전송 데이터의 무결성이 지켜지는 것을 확인할 수 있었다. 이 실험 결과를 통해 다중 통신 상황에서 데이터를 다중으로 처리하는 CMLib의 문제로 인한 데이터의 누락 혹은 오염의 발생이나 본 논문의 설계 및 구현의 오류 없이 데이터의 무결성을 지키며 안정적인 데이터의 처리가 가능함을 확인할 수 있다.

IV. 결론

전자광학장비는 목표물에 대한 감시를 목적으로 하는 장비이기 때문에 관측자에게 끊임 없는 영상과 센서의 정보를 전달해야 하고 제어 명령의 수행 역시 전송 속도 및 처리가 느리다면 사용자에게 매끄러운 기능을 제공하지 못해 문제가 된다.

앞선 실험에서 30Hz로 송신 주기를 설정한 이유는 자연스러운 영상의 전달과 제어 명령의 수행이 가능한 단위로 복미에서 주로 사용되는 텔레비전 수신 방식이며 감시 장비의 카메라에 널리 사용되고 있는 NTSC[9] 표준의 프레임 사이즈인 30fps를 기준으로 선정하였다.

본 논문에서 전술한 점을 해결하기 위해 구현한 통신 미들웨어(CMLib)는 성능 검증을 통해 송신/수신 SW의 오차와 Thread 동기화 그리고 임의의 커널 스케줄링이 불가한 윈도우즈라는 조건 고속의 데이터를 다중으로 처리하면서도 송신/수신 주기의 차이가 거의 나타나지 않음을 확인하였다. 또한 측정 결과 그래프를 통해 제시한 조건 범위 내에서 일정한 간격으로 데이터의 안정적인 통신 효율이 나타남을 확인하였으며, Round Trip Time 실험을 통해 30Hz의 데이터 전송이 가능함을 확인하였다. 또한 고속의 다중 통신 상황에서도 신뢰성 있는 데이터의 처리와 전송이 가능함을 데이터 무결성 실험을 통해 확인할 수 있었다.

향후에는 주로 UDP(User Datagram Protocol) 네트워크 프로토콜을 사용하는 데이터 중심 통신 기술로 네트워크 도메인에서의 참여와 탈퇴가 용이하고 복잡한 분산 환경에서 노드 구성의 유연성과 실시간 노드의 확장성을 극대화할 수 있는 OMG 기구[10]에서 제정한 통신 미들웨어인 DDS(Data Distribution Service)[11] 적용 방안에 대해 검토하여 제안하는 통신 미들웨어(CMLib)와의 성능 비교 및 개선점에 대해 연구할 계획이다.

REFERENCES

- [1] 홍석민, “전자광학 영상센서,” *물리학과 첨단기술*, 11-18쪽, 2006년 1월
- [2] 박신배, 김원제, 이건철, “복합임무 무인수상정의 개발시험평가 및 검증절차에 관한 고찰,” *한국해양공학회지*, 제32권, 제5호, 402-409쪽, 2018년 10월
- [3] 윤제석, 최성혜, 안정훈, “무인기의 자세 변화에 대해 안정된 시선을 유지하는 전자광학 추적장비의 좌표지향 시스템 개발,” *한국항공우주학회 학술발표회 초록집*, 880-881쪽, 2017년 11월
- [4] 최대규, 김대현, 안정훈, “전자광학추적장비의 추적 성능 검증시험,” *대한전기학회 CICS' 18*, 288-289쪽, 2018년 10월
- [5] 윤제석, 유건환, 최병태, 김성수, “전자광학 추적장비의 시선 안정화 제어기 성능 비교,” *대한전기학*

회 CICS' 16, 216-217쪽, 2016년 10월

- [6] 원광은, 안은빈, 김아영, 이홍래, 서광덕, “MMT 프로토콜 기반의 다중쓰레드를 활용한 ARQ 패킷 오류 제어 기법,” *방송공학회논문지*, 제23권, 제5호, 682-692쪽, 2018년 9월
- [7] 임현승, “스칼라 동시 프로그래밍,” *정보과학회지*, 제35권, 제3호, 16-21쪽, 2017년 3월
- [8] 손동오, 심규연, 김철홍, “작업 처리 단위 변화에 따른 GPU 성능과 메모리 접근 시간의 관계 분석,” *스마트미디어저널*, 제4권, 제4호, 56-63쪽, 2015년 12월
- [9] 김형윤, 남해운, 최세영, “SDR 플랫폼을 이용한 NTSC 표준 기반의 TV 송신기 구현,” *한국통신학회논문지*, 제43권, 제12호, 2057-2060쪽, 2018년 12월
- [10] Object Management Group(2019). <https://www.omg.org> (accessed Sept., 20, 2019).
- [11] 윤근재, 최훈, “EchoDDS 통신 미들웨어의 성능 분석,” *정보과학회 컴퓨팅의 실제 논문지*, 제24권, 제7호, 325-336쪽, 2018년 7월
- [12] 서희경, “IOT 및 네트워크 관리 지원을 위한 컴포넌트 아키텍처 개발,” *스마트미디어저널*, 제6권, 제2호, 42-49쪽, 2017년 6월
- [13] 박중선, 조기환, “고속네트워크에서 네트워크 혼잡 상태에 적응적인 UDT 병렬전송 기법,” *스마트미디어저널*, 제2권, 제4호, 51-59쪽, 2013년 12월

저자 소개



배종운(학생회원)

2009년 상명대학교 정보통신공학과 학사 졸업.
2019년 충남대학교 컴퓨터공학과 석사 졸업.

<주관심분야 : 통신 미들웨어, 전자광학장비, 임베디드 시스템>



최훈(정회원)

1983년 서울대학교 컴퓨터공학과 학사 졸업.
1990년 Duke University 전산학과 석사 졸업.
1993년 Duke University 전산학과 박사 졸업.
1996년~현재 충남대학교 컴퓨터공학 교수

<주관심분야 : 모바일 컴퓨팅, 분산 시스템 미들웨어, 운영체제, 컴퓨터 통신>