

RVDB와 대용량 서버 간의 직접 데이터 전송 알고리즘 개발과 시험에 관한 연구 DEVELOPMENT AND TESTS OF THE ALGORITHM FOR DIRECT DATA TRANSMISSION BETWEEN RVDB AND HUGE CAPACITY DATA SERVER

노덕규¹, 오세진^{1*}, 염재환¹, 정동규¹, 오충식¹, 윤영주¹, 김효령¹, Kensuke Ozeki²

¹한국천문연구원, ²Elecs Industry Ltd.

DUK-GYOO ROH¹, SE-JIN OH¹, JAE-HWAN YEOM¹, DONG-KYU JUNG¹, CHUNG-SIK OH¹, YOUNG-JOO YUN¹,
HYO-RYOUNG KIM¹, AND KENSUKE OZEKI²

¹Korea Astronomy and Space Science Institute, Daejeon 305-348, Korea

²Elecs Industry Co. Ltd., Kawasaki 213-0014, Japan

E-mail: dgroh@kasi.re.kr

(Received October 20, 2014; Revised November 08, 2014; Accepted November 18, 2014)

ABSTRACT

This paper describes the development of algorithm for direct data transmission between Raw VLBI Data Buffer (RVDB) and Huge Capacity Data Server (HCDS) operated in Korea-Japan Correlation Center (KJCC). The transmitted data is the VLBI observation data, which is recorded at each radio telescope site, and the data transmitting rate is varying from 1 Gbps, in usual case, upto 8 Gbps. The developed algorithm for data transmission enables the direct data transmission between RVDB and HCDS through 10 Gbps optical network using VLBI Data Interchange Format (VDIF). Proposed method adopts the conventional UDP/IP protocol, but in order to prevent the loss of data during data transmission, the packet error monitoring and data re-transmission functions are newly designed. The VDIF specification and VDIFCP (VDIF Control Protocol) are used for the direct data transmission between RVDB and HCDS. To validate the developed algorithm for data transmission, we conducted the data transmission from RVDB to HCDS, and compared to the transmitted data with the original data bit by bit. We confirmed that the transmitted data is identical to the original data without any loss and it has been recovered well even if there were some packet losses.

Key words: VDIF: Data transmission algorithm: Packet error monitoring: RVDB

1. 서론

한국천문연구원에서 운영하고 있는 대전상관기(Daejeon correlator)는 일본국립천문대와 2006년부터 공동으로 개발하였으며, 2010년부터 한일상관센터에서 한일공동 VLBI관측망(KaVA, KVN¹ and VERA² Array)으로 관측한 VLBI³ 관측데이터의 상관처리에 활용하고 있다(Roh et al., 2008).

대전상관기의 주요 구성품인 동기재생처리장치

(RVDB, Raw VLBI Data Buffer)는 각 관측국에서 기록한 관측 데이터를 임시 저장하거나 동기 재생하는 일종의 대형 디스크 어레이이다. KVN은 하드디스크 기반의 Mark5B 고속기록기를 사용하며, VERA는 자기 테이프 기반의 DIR-2000 고속기록기를 사용하고 있다. 한일상관센터에는 각 미디어를 위해 Mark5B 고속재생기 3대와 VERA-2000 2대를 보유하고 있지만, 서로 다른 규격의 이 장치들로부터 여러 관측국의 데이터를 정확한 시각에 상관처리 장치(VCS⁴, Oh et al., 2009, Oh et al., 2011)로 공급하기가 어렵기 때문에, 먼저 각각의 고속재생기에서 RVDB로 관측데이터를 복사한 다음, RVDB에서 모든 관측국의 데이터를 동시에 재생하도록 하고

* Corresponding author

¹ Korean VLBI Network.

² VLBI Exploring of Radio Astrometry.

³ Very Long Baseline Interferometry.

⁴ VLBI Correlation Subsystem.

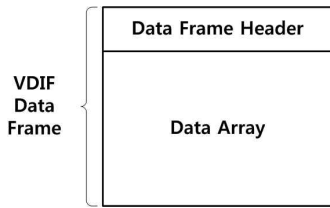


Figure 1. VDIIF data frame structure (Whitney et al., 2010).

있다. 즉, RVDB는 VLBI 관측 데이터를 고속으로 기록할 수 있는 고속기록기, 일정 기간 데이터를 보관하는 대용량 버퍼, 상관처리 시 여러 관측국의 데이터를 동기 재생할 수 있는 고속재생기의 역할을 겸하고 있다. 또 RVDB는 Mark5B나 VERA-2000/DIR-2000 이외에 고속 네트워크에 직접 접속하여 데이터를 송수신할 수 있는 기능도 갖고 있다(Oyama et al., 2008). 각 관측국에서 상관센터까지의 고속 네트워크가 확보될 경우, 이 기능을 활용하여 각 관측국에서 상관센터로 직접 관측 데이터를 전송할 수도 있다.

본 연구에서는 대용량 데이터 서버(HCDS, Huge Capacity Data Server)에서 RVDB 시스템으로 데이터 전송을 수행하는 방법(Yeom et al., 2014)에 이어서, RVDB에서 직접 대용량 데이터 서버로 데이터 전송을 수행하는 방법을 제안한다. 이 방법에서는 대용량 데이터 서버가 RVDB의 입출력 모듈(OCTAVIA)을 경유하지 않고 RVDB의 데이터 저장장치(OCTADDB 또는 OCTADISK)에 직접 접속하여 데이터 전송을 수행하는 알고리즘을 사용하였다. 일반적으로 UDP/IP 프로토콜을 사용하는 광 네트워크에서 흔히 발생하는 데이터 손실을 방지하기 위해, 데이터 패킷을 감시하는 기능과 데이터 재전송 방법을 도입하였다. 이에 따라 RVDB 시스템의 OCTADDB/OCTADISK(Oyama et al., 2008)에 저장된 관측데이터를 대용량 데이터 서버의 스토리지로 손실 없이 전송할 수 있게 되었고, 한일상관센터에서 운영하고 있는 소프트웨어 상관기 또는 VCS Simulator(Yeom et al., 2009) 등에서 상관처리에 활용할 수 있다.

본 논문에서는 데이터 전송 알고리즘의 설계와 성능 시험에 대해 다룬다. 본 논문의 구성은 다음과 같다. 2장에서는 데이터 전송 규격에 대해 기술하고, 3장에서는 제안한 데이터 전송 알고리즘의 제어 시퀀스와 구현한 소프트웨어를 기술하며, 4장에서는 성능 시험과 결과에 대해 고찰하고, 마지막으로 5장에서 결론을 맺는다.

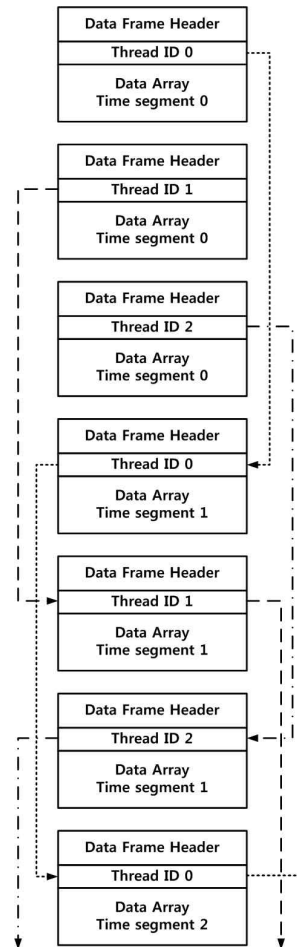


Figure 2. Configuration of data frame by introducing Thread ID (Whitney et al., 2010).

2. 데이터 전송 규격

2.1. VDIIF 규격 및 구조

최근 국내 또는 국가간 초고속 네트워크의 구축이 급속도로 진행되고 있다. 전파천문학 특히 VLBI 관측데이터 획득 및 수집 시스템에서도 초고속 네트워크를 이용하여 실시간 또는 준실시간으로 전송할 수 있도록 관측 데이터를 표현 방식을 표준화하는 새로운 규격인 VDI F5(VLBI Data Interchange Format) 규격을 제정하게 되었으며(Takahashi et al., 2000), 또 e-VLBI(electronic-VLBI)⁶를 위해 네트워크 표준 RTP(Real-time Transport Protocol)/RTCP(Real-time Control Protocol) 프로토콜 기반의 VSI-E7 규격도 제정되었다.

VDIF 규격은 국내 또는 국가간 초고속 네트워크를

⁵ <http://vlbi.org/vdif/>.

⁶ <http://www.haystack.mit.edu/tech/vlbi/evlbi/>.

⁷ <http://vlbi.org/vsi/>.

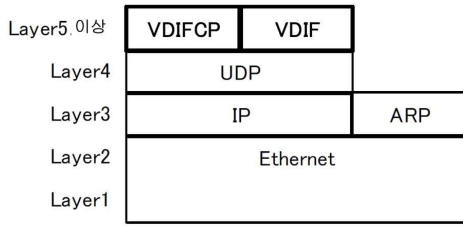


Figure 3. Protocol structure.

통해 VLBI 관측데이터를 전송할 수 있도록 하는 표준이며, e-VLBI를 위한 VSI-E 규격과는 달리, VLBI 관측 데이터를 취급하는 모든 장비 사이에 데이터 전송을 가능하도록 만든 것이 특징이다. 많은 VLBI 관측국에서 이 VDIF 규격을 채택한 장비를 공통으로 갖추게 되면, 데이터 전송이 간결해지고 데이터 처리를 효율적으로 수행할 수 있으므로 대규모의 글로벌 VLBI 네트워크로 확장하기가 쉬워진다.

Figure 1은 VDIF 데이터 프레임 기본 구조를 나타낸다. VDIF 데이터 프레임은 데이터 프레임 헤더와 데이터 어레이(Data array)로 구성된다. 일반적으로는 하나의 네트워크 패킷에 하나의 프레임을 실어 전송하는 구조로 운용된다. 그러나 여러 장비에서 네트워크를 통해 전송되는 데이터를 취급하는 수신시스템은 각각의 장비에서 오는 데이터를 구분해야 하는데, Figure 1의 VDIF 구조에서 데이터 프레임 헤더에 ‘Data Thread’를 도입하면 이 문제를 해결할 수 있다. Figure 2는 ‘Thread ID’를 도입할 때 데이터 프레임이 어떻게 구성되는지 나타낸 것이다. Figure 2에서 네트워크를 통해 수신시스템으로 전송된 VDIF 데이터 프레임에서 Thread ID가 동일한 경우의 데이터 프레임을 찾아서 연결하면 각 장비(송신 시스템)에서 보낸 데이터를 확인할 수 있으며, 여러 장비로부터 전송된 데이터를 각각 독립적으로 구분하여 수신시스템에서 저장할 수 있다. 만약 송신시스템이 오직 하나인 경우에는 Thread ID를 0으로 설정하면 된다.

2.2. VDIF 데이터 프레임 구조

Figure 3은 VDIF 규격에서 사용하는 프로토콜의 구조를 나타낸 것이다. 10GbE 포트로 관측데이터를 전송할 때 VDIF 프로토콜을 사용하고, 관측데이터의 전송제어는 VDIFFCP(VDIF control protocol)를 사용한다. 이는 Layer5 이상에서 정의되며 VDIF 관측데이터를 전송하기 위한 독자적인 프로토콜이다. Layer4이하의 송수신 하드웨어에서는 표준 프로토콜인 Ethernet/IP/UDP를 사용한다.

본 연구의 데이터 전송 알고리즘에서는 Figure 3의 Layer3 이상의 구조를 사용한다. VDIFFCP에는 모두 3종류의 프레임을 정의하는데, 작동 모드에 따라 다음과 같이 다

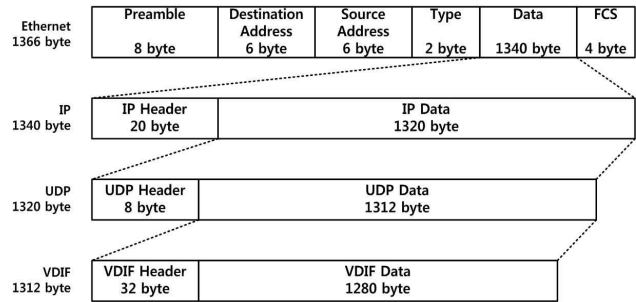


Figure 4. VDIF data frame structure.

른 프레임을 사용한다. MODE1에서는 SR 프레임과 RR 프레임, MODE2는 SR 프레임과 VR 프레임을 사용한다. SR(Send report) 프레임은 관측데이터의 부가적인 정보를 송신측에서 수신측으로 알려주는 프레임이며, RR(Receiver report) 프레임은 관측데이터의 전송품질을 수신측에서 송신측에 알려주는 프레임이다. 또 VR(VDIF request) 프레임은 수신측으로부터 송신측에 관측데이터를 요청하는 프레임이다.

Figure 4에 데이터 전송 알고리즘에서 사용한 데이터 프레임의 구조를 나타내었다. VDIF 데이터는 Figure 3의 UDP/IP 네트워크 층에 조립되어 패킷으로 만들어진다. 전송의 편의를 위해, 관측데이터는 1,280 byte 단위로 분할되어 VDIF data로서 프레임에 적재된다. VDIF 데이터 1,280 byte에 각 층별로 VDIF 헤더 32 byte, UDP 헤더 8 byte, IP 헤더 20 byte, Ethernet 헤더 26 byte가 추가되어 최종 1,366 byte(10,928 bit) 길이의 Ethernet 프레임이 만들어지고, 수신장치로 전송된다. 이러한 프레임은 관측 데이터의 기록 속도가 1,024 Mbps (VSI - 32 MHz)일 때 10µs마다 1개씩 생성되어 초당 100,000 프레임이며, 2,048 Mbps (VSI-64MHz)일 때는 5 µs마다 1개씩 생성되어 초당 200,000 프레임이 된다. 한편 Ethernet 프레임 최저송신간격(inter frame gap)은 96비트이므로 이를 감안하면 최대 2,204.8 Mbps 대역폭이 요구된다.

2.3. VDIF 데이터 프레임

이 연구에서 사용한 VDIF 데이터 프레임의 헤더는 Figure 5와 같이 정의하였다. 표준 VDIF 헤더에서 정의된 필수 항목을 충실히 반영하고 있는 이 외에, 사용자용으로 할당된 구역에 유효비트수, 샘플링, 프레임 분할 정보, 시각부호 유효성 정보 등을 추가하였으며, Table 1에 그 의미를 상세히 나타내었다. 그 밖에 실제 관측데이터가 적재되는 VDIF data 부분은 동일하다.

2.4. VDIFFCP VR 프레임과 SR 프레임

대용량 데이터 서버가 RVDB의 입출력 모듈을 경유하

Table 1. Detailed information of VDIF header

Information	Bit length	Value	Description
Valid bit number	5	0~31	Bit mask 31 : 32bit overall valid 0 : LSB 1bit valid
Sampling	4	0/1	2bit sampling MSB(H), LSB(L) 1 : H frame, 0 : L frame
Split	4	0~5	0 : H/L frame no split 1 : split 32bit by every bit (even bit: H frame, odd bit: L frame) 2 : split 32bit by 2bit (4n~4n+1: H frame, 4n+2~4n+3: L frame) 3 : split 32bit by 4bit (8n~8n+3: H frame, 8n+4~8n+7: L frame) 4 : split 32bit by 8bit (16n~16n+7: H frame, 16n+8~16n+15: L frame) 5 : split 32bit by 16bit (LSB 16bit: H frame, MSB 16bit: L frame)
TV	2	0~3	Time Valid (1: valid, 0: invalid) upper bit: clock code Time Valid lower bit: source timing Time Valid

Table 2. Detailed information of VR frame header

Name	Bit length	Value	Description
version	4	1(fixed)	VDIFCP version
Classification	4	3(fixed)	VDIFCP kinds=VR
VSI port	4	1~4	VSI channel number -when control PC transmit 1~4 temporal number
Reserve	4	ALL 0(fixed)	in future use
Packet Length	16	28(fixed)	VDIFCP length(byte unit)
Reserve	8	ALL 0(fixed)	in future use
Sequence number	24	0~0xFFFFFFFF	Increasing sequence number by every VR transmission 0→1→2...0xFFFFFFFF→0xFFFFFFFF→0
Reserve	8	ALL 0(fixed)	in future use
Mode	4	1 or 2	VDIF mode(1=MODE1, 2=MODE2)
Time code	52	4bit BCD x 13row	observation data time YYYYDDHHMMSS
Reserve	96	ALL 0(fixed)	in future use

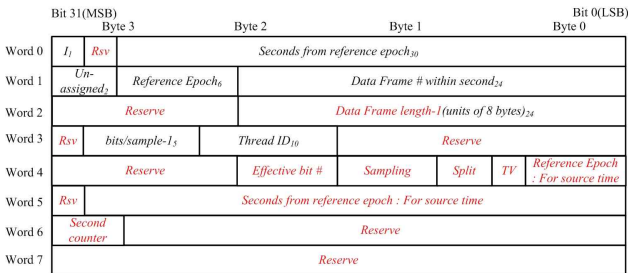


Figure 5. VDIF header specification.

지 않고 RVDB의 데이터 저장장치(OCTADDB 또는 OCTADISK)에 직접 접속하여 데이터 전송을 수행하는 알고리즘은 MODE2에서 구현되며, 2.2절에서 기술한 것과 같이 VDIFCP의 3가지 프레임 중 SR 및 VR 프레임을 이용하고 있다.

수신시스템은 송신시스템으로부터 받고자 하는 VDIF 데이터의 정보를 담은 VR 패킷을 보낸다. VR 프레임 헤더의 구조는 Figure 6에 나타내었으며, 각 필드의 의미를 Table 2에 상세하게 설명하였다. VR 패킷의 전송은 VDIF 데이터 패킷의 송수신과는 분리되어 진행되며, VDIF 데이터 패킷이 전송되지 않더라도 매초 1회의 비율로 지속적으로 발생하게 된다. 수신시스템은 이 VR 프레임 헤더에 나타난 VSI 채널의 번호, 송신요청 일련 번호, VDIF 모드, 전송할 데이터의 시각을 참고하여, 다음 1 PPS에 전송할 데이터를 준비하게 된다.

송신 시스템은 VR 패킷에 따라 준비한 VDIF 데이

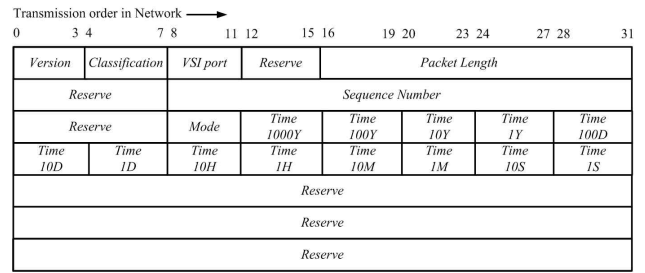


Figure 6. Structure of VR frame header.

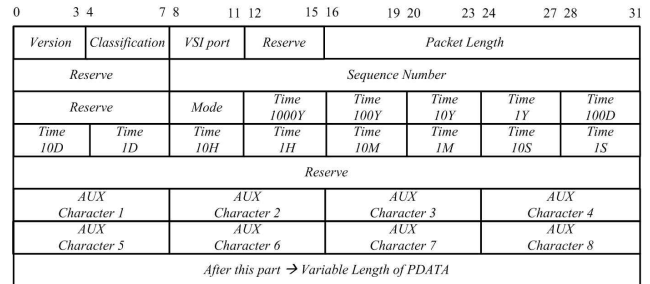


Figure 7. Structure of SR frame header.

터 패킷을 순차적으로 송신하는 한편, SR 패킷을 통해 이 VDIF 데이터 패킷에 대한 정보를 수신시스템으로 전송한다. Figure 7에 SR 패킷의 프레임 헤더 구조를 나타내었으며, 전송되는 VDIF 데이터 패킷의 길이, 전송모드, 시각정보 등을 확인할 수 있다. SR 패킷을 수신한 수신시스템은 SR 패킷의 시각정보를 확인하며 제어 시

퀵스에 따라 VDIF 데이터 패킷의 획득 및 저장을 담당한다.

3. 데이터 전송 소프트웨어

3.1. 전송 제어 시퀀스의 개선

RVDB의 데이터 저장장치(OCTADDB 또는 OCTADISK)에 저장되어 있는 데이터를 RVDB의 입출력 모듈을 사용하지 않고 외부의 데이터 서버로 직접 전송하기 위해서는, 송신시스템에 해당하는 OCTADDB 또는 OCTADISK, 수신시스템에 해당하는 외부 데이터 서버, 그리고 이 두 시스템에 작업을 지시하는 제어 컴퓨터 간의 통신과 이 통신을 적절히 제어하는 알고리즘이 필요하다. 대전상관기에서의 제어 컴퓨터는 송신시스템 및 수신시스템과 분리된 별도의 컴퓨터로 구성되나, 여기서는 외부 데이터 서버가 그 역할을 겸할 수 있다.

이 연구에서 다루는 데이터 전송 제어 시퀀스를 Figure 8에 나타내었다. 최초로 제어 컴퓨터는 수신시스템에게 관측데이터를 전송하게 될 송신시스템의 IP 정보와 파티션 이름(예: ABC)을 알려주고, 전송시스템에게 전송할 데이터가 보관되어 있는 파티션 이름(ABC), 전송할 데이터의 시작시각(t_5), 데이터의 재생속도(1 또는 2 Gbps)를 지시한다. 이때의 시각을 관측데이터의 시각과 동등한 기준으로 나타내면, 시작시각 t_5 보다 5초 앞선 t_0 라고 나타낼 수 있다. 수신시스템에서는 다음 1 PPS부터 매초 VR 패킷을 송신시스템으로 보내어 데이터 송신을 요구한다. $t = t_1, t_2, t_3, t_4, \dots$ 의 각 시점에서 전송되는 VR 패킷에 지정된 시각 $t_0 + 3, t_1 + 3, t_2 + 3, t_3 + 3, \dots$ 은 각각 $t_5 - 2, t_5 - 1, t_5, t_5 + 1, \dots$ 과 같고, 즉 2초 전, 1초 전, 시작시각, 다음 1초, ...의 데이터를 전송하라는 의미이다. 송신시스템은 VR 패킷을 받으면 1초 이내에 지정 시각의 데이터를 읽고 전송할 준비를 해 두었다가 다음 1 PPS 시점부터 VDIF 데이터 패킷을 전송하기 시작한다. 이 때 데이터가 존재하지 않는 시각이 지정되면 보낼 데이터가 없으므로 VDIF 데이터 패킷을 전혀 보내지 않는다. 이러한 데이터 전송 제어 시퀀스에 따라, t_3 시점에 수신한 VR 패킷에 의거하여 시작시각 t_5 의 데이터가 담긴 VDIF 데이터 패킷이 실제 전송되기 시작하는 시점은 t_4 시점이 되며, 이 무렵 다음 1초의 데이터를 전송하라는 $t_3 + 3$ 시각의 VR 패킷이 전달된다. 한편 송신시스템의 안정적인 작동을 담보하기 위하여, 항상 전송 개시시각 보다 2초 이상의 여유를 두고 제어가 진행되도록 설계되어 있기 때문에, 최초 및 두 번째 VR 패킷에 지정된 시각의 데이터가 존재하더라도, 첫 VR 패킷을 받고 2초 간은 데이터 전송을 하지 않는다. 매초 전송되는 VDIF 데이터 패킷의 수는 전송속도가 1 Gbps일 때 10만개, 2 Gbps일 때 20만개이다.

본 연구에서 수신시스템은 대용량 데이터 서버이고, 수신된 VDIF 데이터 패킷의 내용을 파일로 보존하는 것이 목표이다. 파일명은 임의로 지정할 수 있지만, 체계적인 데이터의 관리를 위하여 다음과 같은 파일명 작성 규칙을 따르도록 정했다. 확장자는 'vdif'로 통일하였다.

```
<experiment name>_<station code>_<scan name>[_<auxinfo1>_<auxinfo2>].vdif
```

여기서,

<experiment name> : 관측 코드명, 최대 16문자

<station code> : ASCII 2문자로 나타낸 관측국명

<scan name> : 시작 스캔 명(VEX 파일); 최대 16문자

<auxinfo1> : 부가 정보 1

<auxinfo2> : 부가 정보 2

이며, 부가 정보는 'ccppp' 형식('cc': 사전 등록된 2문자로 된 식별자이며, 예를 들어 VDIF 파일 포맷일 경우 'fd'; 'ppp': 다른 특별히 정의된 문자열 정보)으로 나열한다.

전송되는 VDIF 데이터 패킷의 양이 사용하는 전송 회선의 대역폭에 필적할 정도로 많기 때문에 부적절하게 제작된 시스템(NIC 포함)에서 UDP/IP 패킷의 분실이 발생할 수 있으며, 또 수신시스템의 데이터 기록 능력이 이에 미치지 못할 경우에는 근본적으로 실시간 장비인 OCTADDB 또는 OCTADISK의 VDIF 데이터 패킷을 감당할 수 없게 된다. 이것이 수신시스템으로 대용량 데이터 서버를 활용해야 하는 근본 이유이다. 그러나 제품 카탈로그 상 충분한 기록 성능을 갖는 대용량 데이터 서버라고 하더라도(서버 자체의 관리 프로세스를 포함하여), 다른 작업이 함께 수행되고 있는 비실시간 구동 환경에서는 상황에 따라 UDP/IP 패킷의 분실이 일부 발생하는 경우가 생기며, 이로 인해 수 시간에 걸친 데이터 전송 작업이 무위로 돌아가는 경우가 많았다.

이에 따라, 수신시스템에서 전송된 패킷의 상태를 감시하는 기능을 도입하였다. 패킷 분실 등 오류가 감지된 경우, 즉시 2초 전의 제어 상태로 되돌아가 패킷이 분실된 시각의 데이터를 재전송하도록 요청하고, 이미 발송된 요청에 따라 전송되고 있는 쓸모없는 데이터 패킷의 폐기를 수행하도록 알고리즘을 개선하였다. Figure 8 및 Figure 9에 예시한 것처럼 데이터 시각 t_1 에 오류가 감지된 경우 제어 시각 t_2 에서 2초 전인 제어 시각 t_0 로 되돌려서 t_0 에 대한 데이터 재전송을 요청한다. 마찬가지로 데이터 시각 t_5 에서 오류가 감지되면 제어 시각 t_6 에서 2초 전인 제어 시각 t_4 로 되돌려서 t_4 에 대한 데이터 재전송을 요청하는 방식이다. 이 패킷 오류 감시 기능은 데이터 전송이 진행되는 동안 지속적으로 수행되어 UDP/IP 프로토콜을 사용하는 광 네트워크

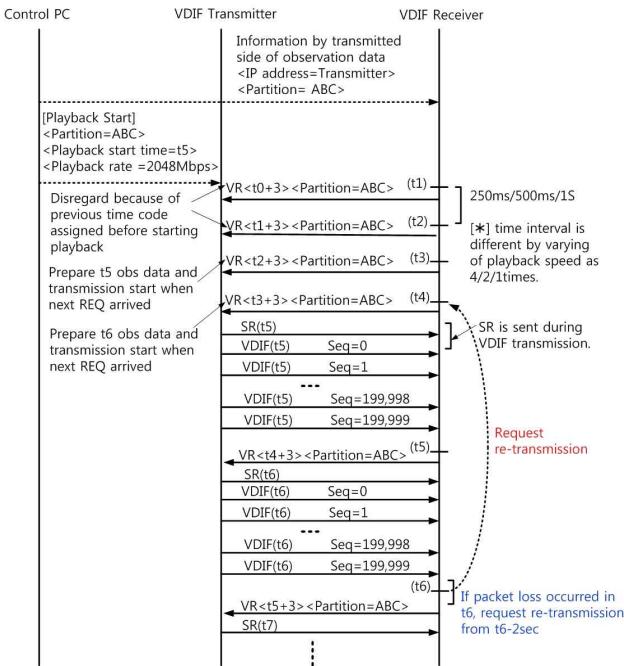


Figure 8. Improved transmission control sequence.

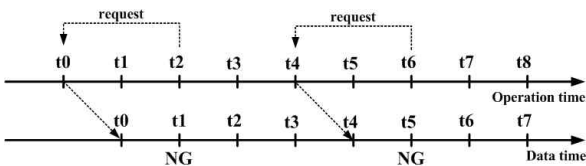


Figure 9. Packet error monitoring and data re-transmission.

크에서 발생할 수 있는 UDP/IP 패킷의 분실 또는 수신 시스템의 부하 변동에 따른 일시적인 데이터 기록 실패로 인한 수신 데이터 손실이 발생하지 않도록 하는 중요한 역할을 담당한다.

3.2. 개선된 전송 제어 시퀀스를 구현한 소프트웨어

본 연구에서 제안한 개선된 전송제어 시퀀스를 적용하여, RVDB(실제로는 그 구성품인 OCTADDB 또는 OCTADISK)에서 대용량 데이터 서버로 대량의 관측 데이터를 손실없이 전송할 수 있는 소프트웨어를 작성하였다. 이 소프트웨어는 기존의 RVDB 제어 소프트웨어와 연동하여 동작하도록 구현되었다. Figure 10에 이 소프트웨어의 흐름도를 간략하게 나타내었다.

수신시스템인 대용량 서버에서 이 소프트웨어가 시작되면, 먼저 RVDB 제어 소프트웨어를 통하여 통신해야 할 송신시스템인 OCTADDB 또는 OCTADISK의 주소, 재생할 관측데이터가 있는 파티션명, 재생 시작시각을 설정한다. 이후 Figure 10에 나타난 것과 같이 매 1초마다 VR 패킷

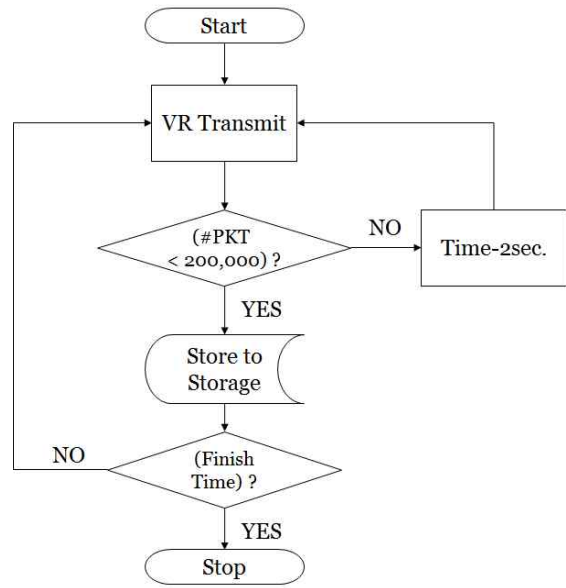


Figure 10. Flow diagram of developed data transmission software.

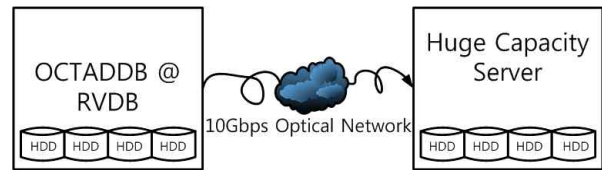


Figure 11. System configuration for VDIF data transmission experiment.

을 송신시스템으로 보내면, 대기하고 있던 송신시스템은 VR 패킷으로부터 재생 시각정보 등을 파악하여 다음 1PPS부터 VDIF 데이터 패킷을 전송해 주는데, 이 때 수신된 패킷을 지정된 파일에 기록하는 한편 이 패킷들의 전송 상태를 감시한다. 데이터 속도 2 Gbps의 경우, 전체 VDIF 데이터 패킷의 개수가 정확히 20만개인 경우에는 서버에 기록된 내용을 확정하고 다음 1초의 데이터 처리를 계속하지만, 만약 패킷에 손실이 발생할 경우에는 서버에 기록된 내용을 무효화하고 그 시각보다 2초 전의 패킷부터 다시 전송하도록 송신시스템에 요구한다. 마지막으로 미리 설정해 둔 재생 종료시각까지의 데이터 전송이 완료되면 프로그램을 종료한다.

일반적으로 실시간 장비가 대용량 데이터 전송을 하는 경우에는 수신측 장비의 상태에 의존하지 않고 데이터 전송을 수행하기 위하여 UDP/IP 방식을 채용하는데, 그 결과 전송 회로 또는 수신시스템의 원인에 의해 전송된 데이터는 흠결이 생길 수 있다. 본 연구에서 개발한 소프트웨어는 송신시스템이 실시간 장비인 관계로 미리 정해진 UDP/IP 방식을 따를 수밖에 없지만, 이미 저장되어 있는 데이터의


```
oper@poli
File Edit View Terminal Tabs Help
#1234, 2013340121924, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 3465600000
#1234, 2013340121925, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 346636800000
#1234, 2013340121926, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 347680000000
#1234, 2013340121927, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 348992000000
#1234, 2013340121928, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 350304000000
#1234, 2013340121929, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 351616000000
#1234, 2013340121930, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 352928000000
#1234, 2013340121931, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 354240000000
#1234, 2013340121932, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 355552000000
UDP receive error!
file pos: exec m: 89791, 35555200000
#1234, 2013340121933, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 355552000000
#1234, 2013340121932, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 355552000000
UDP receive error!
file pos: exec m: 90666, 35555200000
#1234, 2013340121931, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 355552000000
#1234, 2013340121932, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 355552000000
#1234, 2013340121933, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 356864000000
#1234, 2013340121934, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 358176000000
#1234, 2013340121935, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 359488000000
#1234, 2013340121936, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 360800000000
#1234, 2013340121937, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 362112000000
#1234, 2013340121938, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 363424000000
#1234, 2013340121939, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 364736000000
#1234, 2013340121940, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 366048000000
#1234, 2013340121941, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 367360000000
#1234, 2013340121942, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 368672000000
#1234, 2013340121943, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 369984000000
#1234, 2013340121944, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 371296000000
#1234, 2013340121945, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 372608000000
#1234, 2013340121946, 512, 32, 2, ACQ, PDATAVP1, PDATAVP2;
file offset: 373920000000
```

Figure 12. Part of HCDS status terminal for time and transmission during the data receiving from OCTADDB. The square box indicates the re-transmission when transmission error occurred.

전송이라는 특이점을 활용하여 패킷 손실 등 오류 발견 시 해당 시각부터 재전송이라는 방법을 새롭게 도입하여, 대용량의 VLBI 관측 데이터를 RVDB 시스템으로부터 대용량 데이터 서버로 손실없이 복사하는 알고리즘을 구현하였다.

4. 전송실험 및 결과

본 연구에서 개발한 VDIF 데이터 전송 소프트웨어의 성능검증을 위하여, 실제 관측데이터를 대상으로 RVDB의 OCTADDB와 대용량 데이터 서버로 데이터 전송실험을 수행하고 원본 데이터와 전송된 결과를 비교하였다.

```
oper@RVDB
File Edit View Terminal Tabs Help
0000000: edde adab 0000 00d0 1100 4025 0000 0000
0000010: 909a e5a5 14c5 57af 0419 91f6 e515 5e60
0000020: 94f8 a818 9281 6256 f5f5 95bd d09f 5de9
0000030: 1397 a9f9 f67b 55a1 646a 82b5 6abb 1963
0000040: 901a f58e 08ed b756 a8a4 800f 6a88 0115
0000050: 9076 7505 55ac 2566 9e49 6cae 2750 1892
0000060: bf9f 560f 8215 5eac e69d 60cd 1706 5359
0000070: 4a42 006a e3ac 966a 9115 1aa0 15d9 54a6
0000080: be2b f3e5 655a 6396 5748 4a60 7473 9a9f
0000090: a661 216a 0054 154a 59bd 46bb ea1c 98a5
00000a0: 5e7e d096 6097 ceaf f131 5ad1 9e17 67aa
00000b0: 08a8 979c 6c66 b770 aafe a950 197b deb6
00000c0: f5a7 9655 1382 2604 d9dd ca95 fa75 7094
00000d0: 1375 646a 52af c9a9 0239 61aa e6f6 755a
00000e0: 0787 9658 717d 970e 5a6d 6091 1882 6099
00000f0: bc53 a2e8 c995 9502 176a e3d7 aada 9c19
0000100: f666 68a5 b587 c9aa ee51 95d5 fa96 2645
0000110: 7e5f 13d8 eabf 7995 3bd4 3f15 d495 1c65
0000120: 4678 a21a f656 a5d9 5980 90cd 1158 2984
0000130: a22a 5407 cda5 67a2 38f1 577f 2095 5055
0000140: 4792 ab0e eaa9 982a 5469 adf7 1167 9aa2
0000150: 3f8a e59d d0ab 977e eed8 be77 4054 a1a4
0000160: 179d e090 5622 a586 e150 509a ace0 665d
0000170: f8e2 9699 b0bb 1be9 bd72 de44 6afa e56f
0000180: a970 637e 0065 9801 a46c 991a a1a0 5256
0000190: 89a6 66d9 59e2 8812 0a59 6de7 7b20 9e8b
00001a0: c448 6517 2905 4885 6b06 85a2 69a6 a169
00001b0: 03da 9849 5aad 5686 a9fb 7a26 4892 5224
00001c0: 148e 0575 5e19 3571 2268 a949 d489 1084
00001d0: 1aa0 6938 1505 08aa 8a17 509d 0240 1936
00001e0: e1c3 5294 3969 0a52 61a5 9aab 6b0a 9894
00001f0: a8e5 9fae 5b9a 6081 3bd2 79a9 05ad a599
0000200: bf6c 9051 741a a92a a5f1 7d79 dee9 a956
0000210: ea99 b24a 9e7f 4e76 d096 6463 39e2 f9ae
0000220: 6167 664a 666d ba64 6e5d c58a e284 ad43
0000230: a710 ae10 7d36 54de 1a69 79d9 b5e7 a678
0000240: f5dc 7a69 5231 a953 e6a7 eb06 c96f 5764
0000250: a5d7 774a 7e5a 7854 c165 1b9a 0a25 90a9
0000260: 3558 8925 05a4 d13f 2793 70c5 66a2 a985
0000270: 1946 464a b796 a655 950e c356 1396 a26f
0000280: de61 5f69 69f9 bf96 f5db 69b7 7a47 9e11
0000290: 9098 a4a1 2bb2 5882 8191 6628 fe59 4678
00002a0: 578a 9ae1 a476 6a8d f2fd e50e 1484 5519
00002b0: 6480 179a 5592 11af 2182 90e0 c482 8399
00002c0: a05a 0802 0809 109d 4784 a809 9c73 b0b4
00002d0: 7439 667f 1a66 2eae 8a68 2c43 c91d e79e
00002e0: e5fc d3dd 9ae4 9e75 29c6 8504 fa6a e1b6
00002f0: 9598 e5e7 4aa6 a1aa b6ad 5129 aa8a 9086
0000300: 0770 9595 97d7 569e 2801 812c 68f5 a661
0000310: 2512 90e8 e767 0963 f6a6 e50a ad05 ad07
0000320: 0920 9196 0515 2999 1969 a880 b257 fde7
0000330: 5af8 adf6 4fe5 955a 2589 50e6 1da5 c9c6
0000340: 494c 76a5 65fc 776d 1ea6 59ba 0aa1 2224
0000350: 6a87 5016 5908 a654 67e9 a880 1694 7291
0000360: 2198 aa41 ca95 26a2 6e27 00f7 338f 0a45
0000370: da05 6a5e aad2 76c0 de2b 2116 49a5 dc95
0000380: 42a6 0926 2aa2 1859 4cc0 aa64 8302 a56f
540ec00: edde adab 0000 00d0 1100 4025 0000 0000
540ec10: 909a e5a5 14c5 57af 0419 91f6 e515 5e60
540ec20: 94f8 a818 9281 6256 f5f5 95bd d09f 5de9
540ec30: 1397 a9f9 f67b 55a1 646a 82b5 6abb 1963
540ec40: 901a f58e 08ed b756 a8a4 800f 6a88 0115
540ec50: 9076 7505 55ac 2566 9e49 6cae 2750 1892
540ec60: bf9f 560f 8215 5eac e69d 60cd 1706 5359
540ec70: 4a42 006a e3ac 966a 9115 1aa0 15d9 54a6
540ec80: be2b f3e5 655a 6396 5748 4a60 7473 9a9f
540ec90: a661 216a 0054 154a 59bd 46bb ea1c 98a5
540eca0: 5e7e d096 6097 ceaf f131 5ad1 9e17 67aa
540ecb0: 08a8 979c 6c66 b770 aafe a950 197b deb6
540ecc0: f5a7 9655 1382 2604 d9dd ca95 fa75 7094
540ecd0: 1375 646a 52af c9a9 0239 61aa e6f6 755a
540ece0: 0787 9658 717d 970e 5a6d 6091 1882 6099
540ecf0: bc53 a2e8 c995 9502 176a e3d7 aada 9c19
540ed00: f666 68a5 b587 c9aa ee51 95d5 fa96 2645
540ed10: 7e5f 13d8 eabf 7995 3bd4 3f15 d495 1c65
540ed20: 4678 a21a f656 a5d9 5980 90cd 1158 2984
540ed30: a22a 5407 cda5 67a2 38f1 577f 2095 5055
540ed40: 4792 ab0e eaa9 982a 5469 adf7 1167 9aa2
540ed50: 3f8a e59d d0ab 977e eed8 be77 4054 a1a4
540ed60: 179d e090 5622 a586 e150 509a ace0 665d
540ed70: f8e2 9699 b0bb 1be9 bd72 de44 6afa e56f
540ed80: a970 637e 0065 9801 a46c 991a a1a0 5256
540ed90: 89a6 66d9 59e2 8812 0a59 6de7 7b20 9e8b
540eda0: c448 6517 2905 4885 6b06 85a2 69a6 a169
540edb0: 03da 9849 5aad 5686 a9fb 7a26 4892 5224
540edc0: 148e 0575 5e19 3571 2268 a949 d489 1084
540edd0: 1aa0 6938 1505 08aa 8a17 509d 0240 1936
540ede0: e1c3 5294 3969 0a52 61a5 9aab 6b0a 9894
540edf0: a8e5 9fae 5b9a 6081 3bd2 79a9 05ad a599
540ee00: bf6c 9051 741a a92a a5f1 7d79 dee9 a956
540ee10: ea99 b24a 9e7f 4e76 d096 6463 39e2 f9ae
540ee20: 6167 664a 666d ba64 6e5d c58a e284 ad43
540ee30: a710 ae10 7d36 54de 1a69 79d9 b5e7 a678
540ee40: f5dc 7a69 5231 a953 e6a7 eb06 c96f 5764
540ee50: a5d7 774a 7e5a 7854 c165 1b9a 0a25 90a9
540ee60: 3558 8925 05a4 d13f 2793 70c5 66a2 a985
540ee70: 1946 464a b796 a655 950e c356 1396 a26f
540ee80: de61 5f69 69f9 bf96 f5db 69b7 7a47 9e11
540ee90: 9098 a4a1 2bb2 5882 8191 6628 fe59 4678
540eea0: 578a 9ae1 a476 6a8d f2fd e50e 1484 5519
540eeb0: 6480 179a 5592 11af 2182 90e0 c482 8399
540eec0: a05a 0802 0809 109d 4784 a809 9c73 b0b4
540eed0: 7439 667f 1a66 2eae 8a68 2c43 c91d e79e
540eee0: e5fc d3dd 9ae4 9e75 29c6 8504 fa6a e1b6
540eef0: 9598 e5e7 4aa6 a1aa b6ad 5129 aa8a 9086
540ef00: 0770 9595 97d7 569e 2801 812c 68f5 a661
540ef10: 2512 90e8 e767 0963 f6a6 e50a ad05 ad07
540ef20: 0920 9196 0515 2999 1969 a880 b257 fde7
540ef30: 5af8 adf6 4fe5 955a 2589 50e6 1da5 c9c6
540ef40: 494c 76a5 65fc 776d 1ea6 59ba 0aa1 2224
540ef50: 6a87 5016 5908 a654 67e9 a880 1694 7291
540ef60: 2198 aa41 ca95 26a2 6e27 00f7 338f 0a45
540ef70: da05 6a5e aad2 76c0 de2b 2116 49a5 dc95
540ef80: 42a6 0926 2aa2 1859 4cc0 aa64 8302 a56f
```

(a) Original data set. (b) Transmitted data set.

Figure 13. Comparison between original and transmitted data sets (16 hexadecimal output).

데이터 전송 실험을 위한 시스템 구성은 Figure 11과 같다. Figure 11에서 VLBI 관측데이터는 RVDB의 OCTADDB에 저장되어 있으며, 본 연구에서 개발한 전송 프로그램은 대용량 데이터 서버에 설치하였고, 수신 시스템과 제어 컴퓨터의 역할을 동시에 수행하도록 설정하였다. RVDB의 OCTADDB와 대용량 데이터 서버를 연결하는 광모듈 및 광 연결단자는 SFP+와 XFP LR(파장 1,310 nm) 타입으로 장거리 데이터 전송이 가능하다. 실제로 RVDB의 OCTADDB의 기록 및 재생속도는 최대 4.5 Gbps이지만, 본 연구에서 사용한 관측데이터의 데이터 속도는 2048 Mbps이므로, 이 속도를 기준으로 실험을 진행하였다.

Figure 12는 대용량 데이터 서버에서 VDIF 데이터 전송 소프트웨어를 수행할 때 RVDB의 OCTADDB로부터 전송된 VDIF 데이터 패킷의 상태를 표시하는 터미널의 일부를 나타낸 것이다. #1234로 시작되는 각 행에 데이터의 시각정보(년도 YYYY, 연간 통산일 DDD, 시 HH, 분 MM, 초 SS)를 나타내고, 기록속도, 비트 폭, 샘플링 비트 등의 정보를 표시한다. 실험에 사용한 관측데이터는 KVN울산전파망원경에서 1.024 Gbps 속도로 Mark5B 고속기록기로 기록한 것이며, 전송실험에서는 2.048 Gbps 속도로 진행하였다. 관측 시작 시각은

2013y340d00h00m00s이고 전송실험에는 약 8시간의 데이터를 사용하였다. Figure 12에서 □로 표시된 부분은 전송 오류가 발생하여 재전송이 일어난 구간을 나타낸 것이다. 본 연구에서 개발한 소프트웨어에서 실시간으로 UDP/IP 패킷을 검사하면서 제어 시각 12h19m32s(즉, 데이터 시각 12h19m31s)에서 오류를 검출하였으며, 이때 전송 중이던 12h19m31s 및 12h19m32s의 데이터를 무효화하고, 12h19m31s부터 데이터를 다시 보내도록 하는 재전송 요청이 발생한 것을 알 수 있다. 이러한 경우에는 3.1절에서 설명한 바와 같이(Figure 8, 9 참조) 개선된 전송제어 시퀀스에 따라 제어 시각 12h19m33s가 되는 시점에서 제어 시각을 2초 전인 12h19m31s로 되돌린 다음, 전송 작업을 계속 진행한 것이다.

Figure 13은 RVDB의 OCTADDB에 저장된 데이터 원본과 전송실험을 통해 대용량 데이터 서버에 생성된 데이터 파일을 비교하기 위하여, 그 일부를 16진수(hexadecimal)로 비교 표시한 것이다. 전체 시간에 대한 비트 단위의 비교에서도 상이점을 발견할 수 없었으며, RVDB의 OCTADDB에 저장된 관측 데이터가 이러한 전송 방법을 통해 대용량 데이터 서버의 스토리지로 성공적으로 전송되었음을 확인할 수 있었다.

이상의 데이터 전송 실험과 전송 전후의 비교를 통하여, 본 연구에서 제안한 VDIF 데이터 전송 알고리즘이 유효하며, 개발된 소프트웨어가 수신시스템의 상태 변화 등에 따라 일부 전송 오류가 발생하더라도 재전송 요구를 통해 잘 복구할 수 있도록 구현되었음을 확인하였다. 향후에는 광케이블을 이용하는 네트워크의 장점을 활용하여, 관측데이터를 직접 관측국에서 상관센터로 장거리 전송하는 시험을 수행할 예정이다.

5. 결론

본 논문에서는 한일상관센터에서 운영하고 있는 RVDB와 대용량 데이터 서버 사이의 직접 데이터 전송 알고리즘의 개발에 대해 기술하였다. 전송 제어 시퀀스의 개선을 통해 구현된 데이터 전송 알고리즘은 광네트워크를 통하여 VDIF 형식의 관측데이터를 대용량 데이터 서버로 전송할 수 있으며, UDP/IP 통신방법을 채용하고 있지만 데이터 손실을 방지하기 위한 패킷 감시 기능과 재전송 기능을 추가하여 안정적으로 작동하도록 설계하였다. 제안된 알고리즘을 구현하여 개발한 데이터 전송 프로그램을 실제 시스템에 적용하여 전송 실험을 수행한 결과, RVDB의 OCTADDB에 저장된 원본 데이터와 대용량 데이터 서버에서 전송받은 데이터로 작성한 파일은 일치하는 것을 확인하였다. 향후 본 연구에서 개발한 데이터 전송 알고리즘은 KaVA 네트워크에서 e-VLBI 전송에도 유용하게 활용할 수 있을 것으로 기대한다.

REFERENCES

- Oh, S. -J., Roh, D. -G., Yeom, J. -H., et al., 2009, A Study on Development of VLBI Correlation Trial Product, PKAS 24, 1
- Oh, S. -J., Roh, D. -G., Yeom, J. -H., et al., 2011, Performance Evaluation of VLBI Correlation Subsystem Main Product, JISPS 12, 4, 322
- Oyama, T., et al., 2008, The Development and Performance of a New 4 Gbps Disk Recorder and eVLBI Systems Using a 10 GbE Network, 2008 General Meeting of IVS.
- Roh, D. -G., Oh, S. -J., Yeom, J. -H., et al., 2008, 2008 Annual Report on Korea-Japan Joint VLBI Correlator and Receiver Development, Korea Astronomy and Space Science Institute (한국천문연구원)
- Takahashi, F., Kondo, T., & Koyama, Y., 2000, Very Long Baseline Interferometer, Ohmsha
- Whitney, A., Kettenis, M., Phillips, C., & Sekido, M., 2010, VLBI Data Interchange Format (VDIF), 2010 General Meeting of IVS.
- Yeom, J. -H., Roh, D. -G., Oh, S. -J., et al., 2009, Development of Software Correlator for KJJVC, JASS 26, 4, 567
- Yeom, J. -H., Oh, S. -J., Roh, D. -G., et al., 2014, Development of transmission algorithm of VLBI observation data and transmission experiment between Server and RVDB, submitted in JISPS