Regular paper | Journal of the KIECS. pp. 1109-1120, vol. 14, no. 6, Dec. 31. 2019, t. 98, pISSN 1975-8170 | eISSN 2288-2189
http://dx.doi.org/10.13067/JKIECS.2019.14.6.1109

# 광역 네트워크 상의 링 버퍼 기반 대용량 VLBI 데이터 스트림 입출력 구현

송민규** · 김효령* · 강용우* · 제도흥* · 위석오* · 이성모*

# Implementation of Ring Buffer based Massive VLBI Data Stream Input/Output over the Wide Area Network

Min-Gyu Song** · Hyo-Ryung Kim* · Yong-Woo Kang* · Do-Heung Je* · Seog-Oh Wi* · Sung-Mo Lee*

요 약

VLBI 연구 분야에서 관측소와 상관센터 간 네트워크의 품질이 보장된다면 관측 데이터를 각 관측소와 상관센터에 반복 저장하던 기존의 비효율을 극복할 수 있다. 즉, 데이터 분석 작업이 수행되는 상관센터로 데이터 저장을 일원화할 수 있고 이를 통해 데이터 처리의 신속성, 생산성을 향상시킬 수 있다. 이를 구현하기 위해 본 논문에서는 VLBI 관측소에서 생산되는 관측 데이터 스트림을 초고속 네트워크 KREONET을 경유해 상관센터로 직접 전송, 저장하는 원격 기록 시스템을 설계하였다. 이 시스템을 기반으로 데이터 기록을 실시한 결과 패킷 손실이 최소화된 상태에서 관측 데이터가 상관센터의 기록시스템에 안정적으로 저장된 것을 확인하였다.

ABSTRACT

In the field of VLBI, If the quality of the connected network between the VLBI station and the correlation center is ensured, the existing inefficiency of repeatedly storing the observation data in each station and the correlation center can be overcome. In other words, the data center can be unified with the correlation center where data analysis is performed, which can improve data processing speed and productivity. In this paper, we design a massive VLBI data system that directly transmits and stores the observation data stream obtained from the VLBI station to the correlation center via the high - speed network KREONET. Based on this system, VLBI test observations confirmed that the observation data was stored perfectly in the recording system of the correlation center without a single packet loss.

키워드

Massive Scientific Data, Ring Buffer, TCP/IP, Wide Area Network
거대 과학 데이터, 링 버퍼, TCP/IP, 광역 네트워크

## Ⅰ. 서 론

네트워크 및 컴퓨팅 기술의 발전에 힘입어 처리 가능한 데이터 속도 및 용량 역시 기하급수적으로 증가하고 있다. 불과 10여 년 전까지만 하더라도 1Gbps에 그치던 컴퓨터의 네트워크 성능은 10GbE의 대중화

* 한국천문연구원 전파천문본부(mksong, hrkim, byulmaru, dhje, sowi, vsat@kasi.re.kr)
** 교신저자 : 한국천문연구원 전파천문본부
· 접 수 일 : 2019. 10. 22
· 수정완료일 : 2019. 11. 18
· 게재확정일 : 2019. 12. 15
· Received : Oct. 22, 2019, Revised : Nov. 18, 2019, Accepted : Dec. 15, 2019
· Corresponding Author : Min-Gyu Song
Division of Radio Astronomy, Korea Astronomy and Space Science Institute
Email : mksong@kasi.re.kr

속에 10Gbps에 버금가는 성능이 일상적으로 구현되고 있다. 뿐만 아니라 네트워크 백본은 100GbE를 넘어 수 백 Gbps에 육박하고 있다[1]. 이러한 기술의 진보 속에 VLBI 데이터 처리에 있어서도 여러 변화 및 가능성들이 논의되고 있다. 그 한 예로 이더넷 프레임으로의 데이터 포맷 통합을 언급할 수 있다. 이전 방식에서 VLBI는 VSI(: VLBI Standard Interface)라고 불리는 차별화된 인터페이스를 기반으로 데이터 입출력이 이뤄졌다. 하지만 현재는 광 네트워크에서 이더넷 프레임으로 관측 데이터가 송수신 처리되는 형태로 발전하였고 성능, 효율, 확장성 면에서 괄목할 만한 성과를 거두고 있다[2]. 특히 특수 제작된 부품에서 벗어나 시중에 나와 있는 기성품 및 범용 기술을 활용하기에 관측 시스템 운용에 있어 효율성이 증대되고 보다 쉽게 성능 최적화를 달성할 수 있게 되었다[3].

VLBI에서 관측 데이터는 전통적으로 각 사이트에 설치된 기록 시스템에 먼저 저장된 후 차나 비행기 등의 교통수단을 통해 상관센터로 전달되었다. 하지만 기존의 이러한 방식 대신 네트워크 기술의 발전에 힘입어 온라인 상의 관측 데이터 전송이 10여년 전부터 일반적 흐름으로 자리를 잡아가는 추세에 있다[4]. 또한 아직은 낮은 속도로 제한적이지만 각 관측소와 상관센터 간 512Mbps 속도로 실시간 데이터 송수신 및 상관처리가 이뤄지고 있다. 하지만 8Gbps VLBI 관측의 경우에는 스토리지 시스템의 최적화 문제로 인해 KVN 각 관측소에서 얻어진 관측 데이터가 상관센터에서 분석 처리되기까지 상당한 시간 지연이 발생하고 있다. 이로 인해 4채널 VLBI 관측의 비중이 점진적으로 늘어날 것으로 예상되는 가운데 향후 적지 않은 어려움이 예상된다.

KVN 각 관측소와 상관센터 간에 고성능 네트워크가 안정적으로 운영되고 스토리지에 유입되는 관측 데이터의 실시간 기록 및 재생이 구현된다면 상기 언급한 데이터 처리 지연은 최소화될 수 있다. 이를 통해 신뢰성 있는 4채널 관측 수행은 물론 해당 메커니즘을 그대로 적용하여 8채널 32Gbps VLBI 관측 시스템으로 확장시키는 것도 가능하다. 이에 따라 본 논문에서는 원거리의 관측소로부터 전달되는 대용량 관측 데이터 스트림을 상관센터에서 안정적으로 수신, 기록 및 재생할 수 있는 스토리지를 설계하였다. 기존 Mark6 시스템의 경우에는 초고속 데이터의 기록 및 안정적 운용을 위해 SG(: Scatter Gather)라 불리는 별도의 파일 시스템을 채택해 사용하였다. 하지만 이 경우 스토리지를 구성하는 각 디스크에 데이터가 개별적인 바이트 단위로 저장됨에 따라 리눅스 파일로 인식되지 않았고 그 자체로는 상관처리가 불가능한 실정이다. 이를 극복하기 위한 수단으로 FUSE(: Filesystem in Userspace)라 불리는 별도의 소프트웨어 인터페이스를 통해 인식 가능한 파일 형태로 변환하는 불편을 감수해야 했다[3].

우리가 설계한 스토리지에서 네트워크를 통해 유입되는 관측 데이터 스트림은 TCP/IP 소켓 버퍼를 거쳐 시스템 상의 링 버퍼에 수신된 후 RAID 기반 스토리지에 순차적으로 저장된다. 이에 따라 네트워크 성능이 일정하게 유지된다면 입력 스트림에 대한 안정적인 수신은 물론 데이터 기록 과정에 있어서도 현재 문제점으로 지적되고 있는 병목 현상 최소화가 가능하다. 또한 데이터 분석에 있어서도 저장된 데이터를 별도의 추가 가공 없이 파일 형태로 바로 상관처리하는 것이 가능함에 따라 효율성과 생산성을 극대화시킬 수 있다.

이에 따라 본 논문에서는 상기 목표 달성을 위한 핵심 요소로 먼저 고성능 스토리지 설계 및 네트워크 최적화에 대해 논하고자 한다. 설계된 시스템의 성능을 정량적으로 검증하기 위해 실제 시험 관측에 적용하였고 무결성, 패킷 손실 등의 방법으로 관측 데이터의 품질을 정밀 분석하였다. 그리고 실제 상관 처리 및 이미징을 통해 설계된 스토리지의 유용성을 입증하였다.

본 논문은 다음의 순서에 따라 작성되었다. 본 서론에 이어 2장에서 대용량 관측 데이터를 생산하는 KVN의 시스템 얼개 및 이를 처리하기 위한 스토리지 설계에 기술하고자 한다. 3장에서는 KVN 세 관측소와 상관센터를 연결하는 초고속 네트워크 환경에서 UDP 기반의 데이터 스트림 전송 및 성능 평가를 실시하고자 한다. 이어 4장에서 TCP/IP 소켓 기반 링 버퍼를 이용한 데이터 스트림 입출력 구현과 성능 평가, 그리고 스토리지에 기록된 데이터 검증에 서술할 것이고 5장에서 최종 결론을 맺도록 한다.

## II. KVN 시스템 소개 및 스토리지 설계

천체로부터 수신된 아날로그 형태의 전파 신호를 네트워크 상에서 전송 가능한 데이터 스트림으로 출력하는 데이터 소스와 해당 데이터 스트림을 수신 및 저장하는 기록 시스템은 각각 관측소, 상관센터에 위치한다. 각 관측소와 상관센터는 초고속 네트워크로 연결되는데 본 장에서는 이에 기반한 데이터 생성, 전달, 기록에 초점을 맞추어 시스템 개발과 설치에 대해 구체적으로 기술하기로 한다.

### 2.1 KVN의 VLBI 시스템 현황

대용량 관측 데이터를 원격 상관센터에 저장하는 스토리지 설계를 논하기에 앞서 KVN 각 관측소의 VLBI 시스템의 구성에 대해 살펴보기로 한다. 이는 크게 안테나, 수신기를 포함하는 프런트엔드와 샘플러, 데이터 획득 및 기록 시스템을 포괄하는 백엔드로 분류되며 각 시스템 별로 데이터 흐름을 순서대로 요약하면 다음과 같다. 먼저 안테나를 통해 유입된 아날로그 형태의 전파신호는 주파수 대역에 따라 22/43/86/129GHz 네 주파수를 담당하는 수신기 중 하나를 통해 수신되고, BBC를 통해 512MHz 대역폭 크기의 중간 주파수 신호로 변환된다. 나이퀴스트 방정식에 따라 512MHz 대역폭의 아날로그 신호가 디지털 변환되기 위해서는 최소 1024Msps, 2 비트로 양자화되어야 하는데 해당 기능을 수행하는 샘플러를 통해 2Gbps 속도로 디지털 신호가 출력된다. 이는 하나의 수신기로부터 출력되는 데이터 양으로 KVN은 4대의 수신기를 이용해 동시 관측이 이뤄지기 때문에 샘플러로부터 출력되는 데이터의 총합은 8Gbps이다[5].

기존의 VLBI 기록 시스템은 1Gbps 기록 속도의 한계로 인해 수신기 4대로부터 출력되는 신호 중 하나에 대해서만 기록이 가능했고 대역폭 역시 256MHz로 제한적이었다. 하지만 2012년 성능이 대폭 개선된 Mark6가 출시됨에 따라 4대의 수신기 시스템으로부터 출력되는 512MHz 대역의 중간주파수 신호를 그대로 동시 기록하는 것이 가능하게 되었다. 이에 대한 효율적인 데이터 전달 및 기록을 위해 4대의 샘플러로부터 각각 출력되는 2Gbps 디지털 신호는 FILA10G, OCTAD 등의 광전송장치를 통해 8Gbps 속도의 이더넷 프레임으로 합성 변환되고 10GbE NIC가 탑재된 Mark6에 전송되어 기록된다. 그림 1은 이러한 데이터 흐름을 나타내는 KVN의 간략화된 시스템 구성도를 보여주고 있다[6].

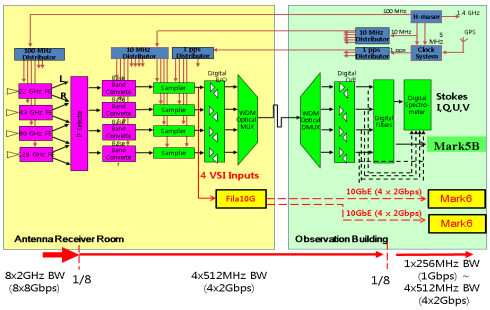



그림 1. KVN의 시스템 구성도 및 VLBI 데이터 흐름
Fig. 1 KVN's system diagram and path for VLBI data

### 2.2 초고속 스토리지 설계

시중에 출시된 스토리지를 보면 Dell EMC, HPE, IBM와 같은 메이저 업체부터 Oracle, Pure Storage 등의 중견업체에 이르기까지 초고속 데이터 입출력과 안정성을 주요 사양으로 내세우며 차별화하고 있다. 이들 제품은 일반 사용자 입장에서는 충분한 데이터 입출력 특성을 제공하며 사용에 있어서 별다른 문제가 없다[7]. 하지만 기본적으로 비정형 데이터 처리에 초점이 맞춰져 있는 관계로 VLBI, HEP(High Energy Physics) 등 수 ~ 수십 Gbps에 달하는 빅 데이터 스트림을 처리하는 분야에서 그대로 사용하기에는 한계가 있다[8]. 기존 스토리지가 성능 면에서 한계에 부딪힐 수밖에 없는 것은 구조적 측면에서 하드웨어 자체보다는 소프트웨어에서 원인이 있다. 스토리지 도입 시 기본 탑재된 파일 시스템 또는 Lustre 등 오픈소스의 경우 대부분 작은 용량의 비정형 데이터 처리에 최적화되어 있다. 따라서 VLBI에서와 같이 10Gbps에 버금가는 속도의 데이터를 저장할 경우 패킷 유실과 성능 불안정이 불가피하다[6].

이에 따라 본 논문에서는 일반적인 파일 입출력에 최적화된 업체 제공의 소프트웨어 사용을 지양하고 최대한 단순화된 구조로 시스템을 설계하였다. 하드디스크 어레이의 경우 단순히 XFS 파일 시스템으로 포맷하였고, 시스템 내에서 입출력 성능 극대화를 위해 메모리 버퍼링을 활용하였다. 이를 구현하기 위한

세부 기술로는 초고속 데이터 수신 분야에서 효율성이 입증된 링 버퍼를 채택하였다. 링 버퍼는 메모리의 효율적 활용을 위해 제시된 개념으로서 메모리의 시작과 끝이 서로 연결된다. 데이터 입출력에 비례하여 입출력 포인터는 중단 없이 실시간으로 전진하며 적절한 속도로 버퍼링이 이뤄질 경우 안정적인 입출력 구현이 가능하다. SDRAM은 CPU에서 직접 접근이 가능한 주 기억 장치로서 3200MT/s (MegaTransfer per second)에 육박하는 입출력 특성을 갖는다. 따라서 시스템 내에 입력된 데이터가 최종 목적지인 디스크에 기록되기 직전, 안정적으로 버퍼링할 수 있는 최적의 수단이라 할 수 있다[9].

링 버퍼를 구현하기 위해 현재 전 세계적으로 가장 널리 쓰이는 방식은 PF_RING 라이브러리를 사용하는 것이다. PF_RING 라이브러리는 링 버퍼를 보다 용이하게 구현하고 효과적으로 사용할 수 있도록 이탈리아 개발자 Luca Deri가 고안한 것으로 내부의 로컬 네트워크 상에서 각 프레임 단위에 대해 개별적으로 반응한다[10]. 이러한 특성으로 인해 확장성 면에서 제약이 발생하는데 그 원인 및 해결 방안을 기술하면 다음과 같다. PF_RING 방식에서 시스템이 처리해야 할 데이터는 MAC 프레임으로서 그 자체만으로는 TCP(: Transmission Control Protocol), UDP(: User Datagram Protocol) 등 프로토콜 적용은 물론 흐름제어를 통한 패킷 처리가 불가능하다[11]. MAC/IP/TCP/UDP 등 사용자가 직접적으로 취급하지 않는 헤더 정보가 함께 수신됨에 따라 페이로드 검출을 위해 일일이 전단의 헤더를 별도로 제거하는 절차가 필요할 수 있다. 무엇보다 그 자체만으로는 데이터 전송을 위한 프로토콜을 지원하지 않기 때문에 RF_RING은 네트워크 안정성이 보장된 LAN(Local Area Network) 영역으로 사용이 제한된다. 전달의 신뢰성 및 전송 범위 확대에 있어 이는 크나큰 손실이라 할 수 있다.

이에 따라 본 논문에서는 외부 네트워크로부터 입력되는 패킷을 블로킹 방식으로 처리하여 컨텍스트 스위칭 발생을 최소화한 것은 물론 네트워크에 연결되어 있다면 어디에서라도 관측 데이터를 기록할 수 있도록 시스템을 설계하였다. 이를 위해 기존의 PF_RING 라이브러리 대신 TCP/IP 소켓 인터페이스를 기반으로 링 버퍼를 설계하였고 소켓 버퍼에 수신

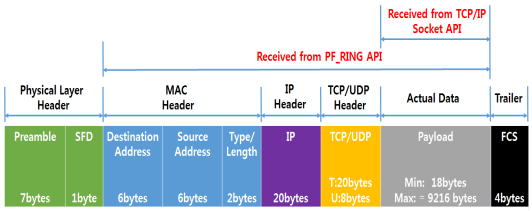



그림 2. 이더넷 프레임 헤더 정보 및 서로 다른 데이터를 반환하는 PF_RING과 TCP/IP 소켓 API

Fig. 2 Different header size of ethernet frame returned data from PF_RING and TCP/IP socket API

된 패킷이 링 버퍼로 복사될 수 있도록 하였다. TCP/IP 네트워크 상에서 데이터는 IP 기반으로 라우팅 되며 소켓 주소에 명시된 IP 주소, 포트 번호에 따라 목적지 시스템의 애플리케이션으로 전달된다. 따라서 데이터 전송에 TCP/IP 소켓 인터페이스를 이용할 경우 애플리케이션에서 데이터 수신 후 즉각적인 활용은 물론 TCP를 기반으로 흐름 제어, 재전송을 구현하는 것이 용이하다. 목적지 MAC 주소(Destination MAC Address)부터 말단에 위치한 FCS(: Frame Check Sequence)에 이르기까지 MAC 프레임 자체를 다루는 PF_RING과 달리 TCP/IP 소켓 인터페이스를 거쳐 수신된 패킷은 MAC, TCP/UDP, IP 헤더가 제거된 상태로 최종 애플리케이션에 전달된다.

그림 2는 이더넷 프레임의 구조 및 헤더 정보를 나타낸 것으로 PF_RING과 TCP/IP 소켓 두 가지 방법으로 링 버퍼를 구동할 경우 사용자에게 최종 전달되는 데이터의 차이를 보여주고 있다. TCP/IP 소켓 기반으로 링 버퍼를 구현하는 경우 MAC/IP/TCP(UDP) 헤더는 시스템 커널 상에서 처리되고 사용자 애플리케이션에는 실제 필요한 페이로드가 수신된다. 하지만 PF_RING 기반 링 버퍼 방식에서는 MAC Header의 Destination Address부터 Payload를 포함하는 MAC 프레임이 전달된다. 따라서 애플리케이션에 필요한 페이로드를 추출하는 과정이 수반되어야 한다. 이 밖에 커널에서 제공되는 TCP 알고리즘을 활용하지 않기 때문에 신뢰성 있는 데이터 송수신은 사용자가 애플리케이션 내에 별도로 구현해야 한다. 이는 데이터를 주고받는 두 시스템이 서로 다른 네트워크에 연결되어 있을 경우 TCP/IP 소켓 기반의 링 버퍼가 보다

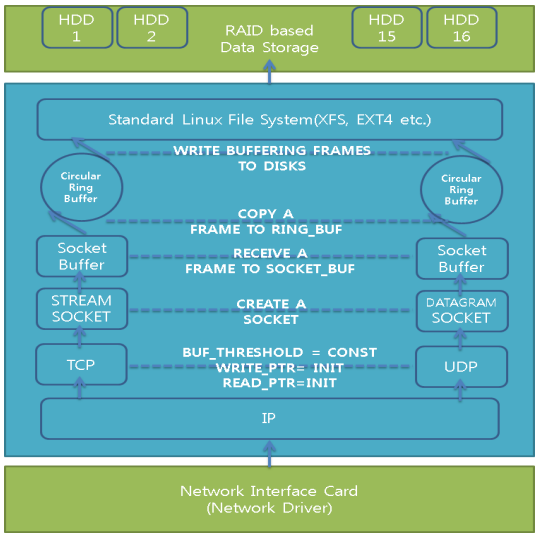



그림 3. 소켓기반 링 버퍼 원리 및 이를 이용한 디스크 상의 데이터 저장
Fig. 3 Operation principle of TCP/IP socket API based ring buffer and data storage

안정적인 전송 옵션을 제공할 수 있음을 의미한다.

상기 2.1절에서 언급하였듯이 수신기 시스템으로부터 획득한 광대역 관측 신호는 샘플러, FILA10G/OCTAD를 거쳐 네트워크 상에서 전송 가능한 UDP 프레임으로 출력된다. 기록시스템의 NIC에 입력된 해당 데이터는 recv, recvfrom 등의 소켓 API(: Application Programming Interface) 함수를 통해 소켓 버퍼에 저장된다. 소켓 버퍼는 외부로부터 매 순간 실시간 입력되는 패킷을 임시 저장하기 위한 메모리 영역이다. 따라서 기존에 소켓 버퍼에 저장되어 있는 패킷은 새로 입력되는 패킷으로 덮여 쓰여지기 전에 링 버퍼에 복사될 필요가 있으며 memcpy 함수 호출을 통해 이를 구현할 수 있다. 시스템에 실시간으로 입력되는 패킷은 상기 과정의 반복을 통해 링 버퍼에 저장된다. 하지만 링 버퍼에 너무 많은 패킷이 누적될 경우 디스크 기록 과정에서 손실이 발생할 수 있다. 따라서 버퍼링되는 패킷 용량이 일정 크기 이상이 되면 write 함수를 통해 디스크에 저장되도록 하였다.

링 버퍼를 이용한 데이터 입출력 구현에 있어 숙지해야 할 주요 사항으로 입력 및 쓰기 포인터가 있다. 쓰기 포인터는 네트워크 인터페이스로 입력되어 링 버퍼에 쓰여지는 데이터의 주소를 추적하고 읽기 포인터는 최종 목적지인 디스크 저장을 위해 링 버퍼로부터 읽혀지는 데이터의 주소를 추적한다. 링 버퍼 상의 데이터 입출력 과정에서 두 포인터는 점진적으로 증가하며 두 포인터 간의 차이가 링 버퍼 상에서 실제 버퍼링되는 데이터 용량에 해당한다[12]. 지금까지 링 버퍼의 동작 원리는 물론 UDP 프레임 수신부터 메모리 상의 버퍼링 및 파일 기록에 이르기까지의 메커니즘에 대해 기술하였다. 이에 대한 시스템 구성을 간략화하여 도식화하면 그림 3과 같다.

설치한 스토리지의 제원 및 주요 컴포넌트 사양은 그림 4에 요약 정리하였고 이를 간략히 기술하면 다음과 같다. 먼저, CPU는 2개의 소켓을 통해 듀얼 CPU가 장착되어 있고 각 CPU에서는 4개의 물리 코어와 하이퍼 쓰레딩이 지원된다. 따라서 논리적으로 활용 가능한 전체 CPU 코어의 개수는 16개이다. 메모리와 NIC는 각각 DDR4 SDRAM 64GB, X520칩셋 기반이 장착되어 있고 16개의 1TB 디스크는 레이드 컨트롤러 PERC H730에 연결되어 RAID0 기반 16TB 용량의 가상 디스크로 마운트되어 있다. 따라서 기록된 데이터는 일반 리눅스 파일로 바로 인식되며 별도의 가공 없이 즉시 분석 활용이 가능하다. RAID, NIC 등 다수의 인터페이스를 통해 대용량 데이터가 입출력되는 과정에서 고속의 인터럽트가 발생함에 따라 세심한 시스템 튜닝이 요구된다. 이를 위해 고속의 입출력을 수행하는 입출력 디바이스는 모두 0번 CPU 노드에서 작동하도록 설정하였고 최대한의 성능을 이끌어낼 수 있도록 numactl을 이용해 애플리케이션을 실행하였다.

| Items | Description |
|---|---|
| CPU | 2CPU / Intel Xeon E5-2623 v3 3.0GHz, 10M Cache, 8.00GT/s QPI, Turbo, HT, 4C/8T (105W) |
| Memory | 64GB (4ea * 16GB) RDIMM, 2133MT/s, Dual Rank, x4 Data Width |
| Raid Controller | PERC H730 1GB |
| Storage Capacity | 16TB (16EA * 1TB) 7.2K RPM SATA 6Gbps 3.5in Hot-plug Hard Drive, 13G |
| NIC | Intel X520 DP 10Gb DA/SFP+ Server Adapter |
| Transceiver | 1ea * SFP+, SR, Optical Transceiver, Intel, 10Gb-1Gb |

그림 4. 대용량 VLBI 데이터 스트림 기록을 위한 모의 데이터 저장 시스템의 하드웨어 사양
Fig. 4 Hardware specification of experimental storage system for massive VLBI data stream

## III. 초고속 네트워크를 경유한 대용량 데이터 스트림 전송 실험

VLBI에서 각 관측소에 저장된 관측 데이터는 분석 처리를 위해 상관센터에 또다시 저장되어야 한다. 이로 인해 데이터 처리 과정에서 비효율이 발생하였고 관측 후 실제 상관처리까지 적지 않은 시간이 소요되고 있다. 이를 해결하기 위한 방법에는 원격 상관과 원격 기록 두 가지가 있다. 본 장에서는 이 중 원격 기록을 구현하기 위한 네트워크 환경, 엔드 단 시스템 설계에 대해 기술하고자 한다.

### 3.1 실험 네트워크 환경

관측소에서 획득한 관측 데이터를 실제 분석 처리가 이뤄지는 원격의 상관센터에 바로 저장하기 위해서는 데이터를 기록하는 스토리지의 성능은 물론 관측소 - 상관센터 간 안정적인 전송을 보장하는 고성능 네트워크가 함께 요구된다. KVN의 광대역 VLBI 관측에서 8Gbps 관측 데이터를 생성 및 전송하는 데이터 소스로서의 역할은 각 관측소 망원경동의 수신기실 내에 설치되어 있는 FILA10G가 담당한다. 종래에 FILA10G로부터 출력되는 관측 데이터는 로컬 네트워크 상의 Mark6 시스템에 저장되었다. 이를 위해 FILA10G 출력단과 연결되는 광점퍼코드가 수신기실 내의 광분배함을 거쳐 관측기기실 내의 광분배함과 직결되고 그 후단에서 해당 기록시스템의 입력단으로 네트워크가 구성되었다. 하지만 기록시스템이 천문대 내부가 아닌 외부에 위치해있고 네트워크를 통해 접근 가능하다면 FILA10G의 출력 포트는 공용 네트워크 스위치에 연결될 필요가 있다. 이에 따라 FILA10G로부터 관측기기실 내의 FDF로 연장되는 싱글모드 광점퍼코드를 KREONET 회신이 업링크되어 있는 네트워크 스위치에 연결하였다. 이러한 경로 설정을 통해 FILA10G로부터 출력되는 UDP 프레임 형태의 관측 데이터는 KREONET 기반 e-KVN 백본을 경유하여 상관센터 내의 스토리지에 직접 전달이 가능하다. 원격 데이터가 전송되는 백본에 해당하는 KREONET 상의 데이터 흐름 및 그 양단에 해당하는 KVN연세전파천문대, 대전상관센터 내부의 네트워크 설정 현황을 나타내면 그림 5와 같다.

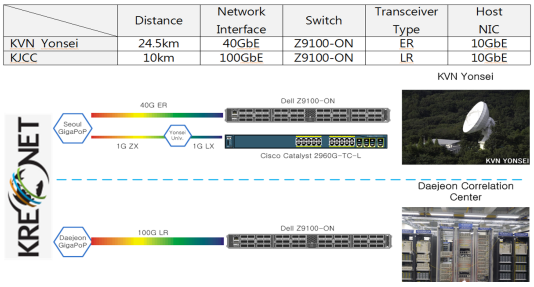



그림 5. KVN연세전파천문대와 대전상관센터 간 초고속 연결을 위한 광 네트워크 인터페이스
Fig. 5 Optical network interface for the high speed connection between KVN yonsei station and daejeon correlation center

### 3.2 UDP 기반 관측 데이터 스트림 전송

위로부터 알 수 있듯이 KVN연세전파천문대와 대전상관센터는 각각 인접한 서울, 대전 지역망센터에 40GbE, 100GbE 인터페이스로 연결되어 있다. KREONET 백본의 핵심에 해당하는 서울, 대전 지역망센터 간 네트워크 회선은 100GbE 2코어로 구성된다. 따라서 KVN연세 관측소와 상관센터 구간에서 전송 가능한 최대 네트워크 성능은 40Gbps이다. 하지만 종단에 연결되어 있는 컴퓨터 시스템의 네트워크 인터페이스가 10GbE이기 때문에 실제 송수신 가능한 데이터 속도는 10Gbps로 한정된다. 전송하는 데이터양에 비해 네트워크 백본의 성능이 월등하면 송수신 간 시스템 간 네트워크 회선이 길더라도 손실이 최소화된, 안정적인 데이터 전송 효과를 얻을 수 있다. 관련해서 KVN연세 사이트, 상관센터의 시스템 간 측정된 데이터 전송 성능을 도시하면 그림 6과 같다. 그림 6의 그래프로부터 KVN연세전파천문대와 대전상관센터 구간의 네트워크 전송 성능이 9.8Gbps 수준이고 안정적으로 유지되는 것을 알 수 있다. 이와 더불어 본 논문에서는 초고속 네트워크를 경유한 관측 데이터 스트림 전송의 실효성을 검증하였다. 이를 위해 vdiftimeUDP 프로그램을 이용하여 FILA10G로부터 출력되는 8,224 바이트 크기의 VDIF 프레임이 상관센터 내의 스토리지에 안정적으로 수신되는지를 모니터링하였다. vdiftimeUDP 프로그램은 FILA10G로부터 출력되는 VDIF 프레임이 수신 시스템에 손실없이 안정적으로 전송되는지를 검증하기 위해 2014년 Jan Wagner에 의해 개발되었다. 이는 VDIF 헤더 내의

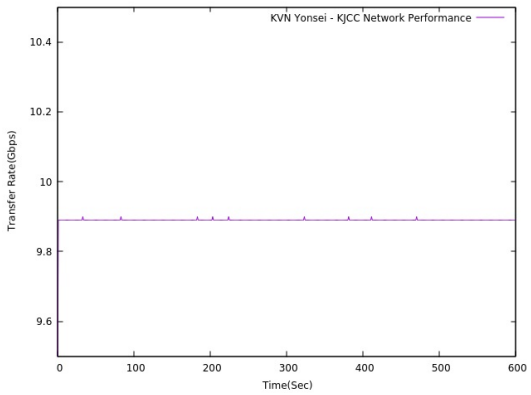



그림 6. KVN연세 사이트와 상관센터 간 네트워크 전송 성능
Fig. 6 Network transmission performance between KVN yonsei station and daejeon correlation center

프레임 번호가 매 초당 0으로 초기화되어 순차적으로 증가하는 특성을 이용해 프레임 손실 여부를 판별한다.

KVN에서 4채널 수신기를 이용한 VLBI 관측이 프레임 손실 없이 완벽하게 이뤄질 경우 8,224 바이트 크기의 VDIF 프레임은 정확히 125,000개 수신된다. 하지만 네트워크 및 시스템 불안정으로 인해 프레임이 유실되면 이보다 적은 개수의 프레임이 수신될 것이다. 그림 7은 이에 대한 결과를 보여주고 있다. 그림 7에서도 확인할 수 있듯이 실험이 진행된 10분 동안 초기 단 한 번의 손실만 발생했을 뿐 관측 데이터 스트림은 KVN연세전파천문대 FILA10G에서 상관센터의 스토리지로 안정적으로 전송되었다. 이는 KREONET 기반 e-KVN이 VLBI 관측 데이터가 생성되는 KVN연세 관측소에서 분석 처리가 이뤄지는 대전상관센터까지 실시간 관측 데이터 스트림 전송 구현에 필요한 성능을 이미 확보하고 있음을 보여준다. 나아가 안정적인 네트워크 환경과 상관센터에 고성능의 스토리지가 보장된다면 관측 데이터가 손실 없이 원격의 스토리지에 기록될 수 있음을 보여주고 있다.

## Ⅳ. 링 버퍼를 이용한 데이터 스트림 입출력 성능 평가 및 데이터 기록 구현

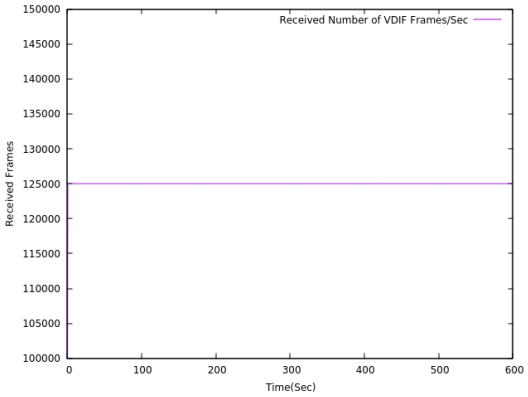



그림 7. 2048MHz x 4Channel 용량의 대용량 VDIF 프레임을 이용한 KVN연세전파천문대 대전상관센터 간 데이터 전송 실험
Fig. 7 Data transmission experiment between KVN yonsei and daejeon correlation center using 2048MHz x 4 Channels of massive VDIF frames

기존의 방식에서 외부로부터 입력되는 데이터는 시작과 끝이 존재하는 한정된 용량의 소켓 버퍼에 저장되었다. 따라서 버퍼가 충만하게 되면 메모리 상의 버퍼링이 불가하게 되었고 이는 데이터 손실을 발생시키는 주 요인 중 하나였다. 본 장에서는 종단 없는 링 버퍼 기반 대용량 스트림 입출력 수행 및 스토리지 상의 데이터 기록 방안에 대해 기술하고자 한다.

### 4.1 링 버퍼 성능 평가

대용량 데이터 캡쳐 및 수신을 위한 수단으로 본 논문에서는 TCP/IP 소켓 기반의 링 버퍼를 사용하였다. 관련해서 소켓 기반 링 버퍼의 입출력 성능을 현재 널리 쓰이고 있는 PF_RING 기반 링 버퍼와 비교하였다. 객관적 성능 분석을 위해 프로그램 내에서 오직 입출력 패킷을 처리하는 함수를 TCP/IP 소켓, PF_RING으로 차별화시켰고 각 경우에 대한 성능 측정을 수행하였다. 각 방식에서 패킷 입출력 구현을 위해 사용된 함수는 위의 표 1에 요약되어 있다.

표 1. 데이터 입출력 처리를 위한 PF_RING과 TCP/IP 소켓 API
Table 1. PF_RING and TCP/IP Socket API for data Input/Output processing

| Interface | PF_RING | TCP/IP Socket |
|---|---|---|
| System Call | pfring_open() | socket() |
| | pfring_set_application_name() | |
| | pfring_get_bound_device_address() | bind() |
| | pfring_enable_ring() | |
| | pfring_recv() | recvfrom() |
| | pfring_close() | |

PF_RING 방식의 경우 PF_RING 소켓 초기화, 함수 호출에 필요한 구조체 핸들 획득을 비롯한 애플리케이션 이름 지정, 링 버퍼 활성화, 디바이스 획득 및 해제 등의 작업이 필요하다. 반면 TCP/IP 소켓 인터페이스의 경우는 일반적인 TCP/IP 통신에서와 동일하게 소켓 개설, 구동하고자 하는 프로그램과 소켓 주소의 결합 후 recvfrom 함수 호출을 통해 외부에서 입력되는 UDP 패킷을 수신하는 형태이다. 본 논문에서는 로컬 네트워크 환경에서 위 테이블에 명시된 함수 호출을 기반으로 FILA10G로부터 전송된 VDIF 프레임을 스토리지에 수신 및 저장하였고, 해당 결과를 간략히 도시하면 그림 8과 같다.

이러한 성능 평가는 FILA10G로부터 8.274Gbps 속도로 수신되는 UDP 프레임을 1TB 용량 16개 디스크로 구성된 스토리지에 15,500초 동안 수신 및 기록하는 방식으로 진행되었다. 실험은 각 방식에서 두 번에 걸쳐 수행하였는데, TCP/IP 소켓 기반 링 버퍼 방식의 경우 두 번 모두 프레임 손실이 발생하지 않았다. 하지만 PF_RING 방식은 첫 번째 실험에서는 프레임 손실이 나타나지 않았으나 두 번째에서는 5번의 프레임 손실이 발견되었다. 유실된 프레임 개수는 359개로서 스토리지에 저장되는 VDIF 프레임이 8224 바이트

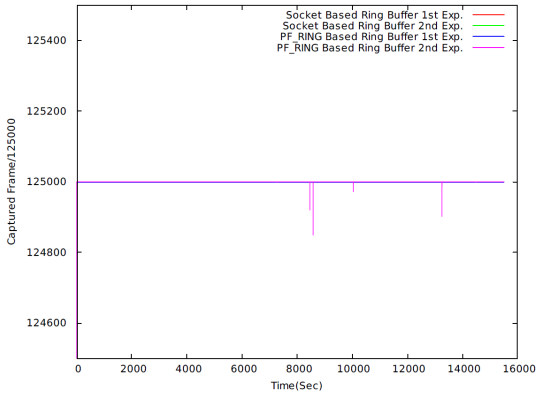



그림 8. PF_RING 기반 링 버퍼와 소켓 기반 링 버퍼를 이용한 데이터 전송 및 성능 평가
Fig. 8 Data transfer using PF_RING and TCP/IP socket based ring buffer and performance evaluation

단위라는 점을 감안했을 때 실질적으로 손실된 데이터 용량은 3MB 내외에 해당한다. 이는 전체 스토리지 용량 면에서 미미한 수준으로서, 본 실험을 통해 TCP/IP 소켓 인터페이스에 기반한 링 버퍼가 기존의 PF_RING 방식에 못지않은 성능과 안정성을 보장함을 알 수 있다.

### 4.2 데이터 분석 및 기록 성능 검증

링 버퍼를 입출력된 스트림은 후단에 위치한 스토리지에 4시간 18분 동안 저장되었다. FILA10G에서 8.274Gbps 속도로 생성되어 네트워크를 가로질러 전송된 데이터 스트림은 대전상관선터에 설치된 스토리지의 내부에서 50 바이트(MAC 헤더: 14바이트, IP 헤더: 20바이트, UDP 헤더: 8바이트)의 헤더 정보가 제거된 상태에서 기록된다. 따라서 최종 기록 단계에서 데이터 속도는 8.224Gbps이며 RAID0 기반 16 x 1TB 디스크를 가득 채우는데 소요되는 시간은 대략 16000초가 소요된다. 링 버퍼 상의 데이터 유실 개수, 디스크 상의 기록 누락 등을 모니터링하는 스크립트의 최종 분석 결과를 얻기 위해 데이터 기록이 이보다 대략 1분 짧은 15500초 동안 수행되도록 설정하였다. 그림 9는 해당 실험의 진행 내역 일부와 결과를 나타내는 것으로 6월 10일 오전 9시 30분경부터 15500초 동안 8.274Gbps 전송율로 입력되는 데이터를 링 버퍼 상에서 처리하는 단 한 번의 패킷 유실도 발생하지 않았다. 후단에 위치한 디스크 상에서 데이터 기록이

완벽하게 이뤄졌다면 기록된 데이터 총량은 다음과 같아야 한다.

Recorded Data Capacity = Input Stream Number X Data Capacity per Stream

```
Time      Frame_Num/Sec          Total_Frame          Buf_Stream_Size      Data_Size          Packet_Loss
          Total_LFrame           Buf_Zone
13858200 85682/125000            85682                704648768            699451200          39318
          39318                  5189344
13858201 125000/125000           210682               1732648768           1718651520         0
          39318                  10025056
13858202 125000/125000           335682               2760648768           2747844000         0
          39318                  10049728
13858203 125000/125000           460682               3788648768           3777036480         0
          39318                  10057952
13858204 125000/125000           585682               4816648768           4806228960         0
          39318                  10016832
13858205 125000/125000           710682               5844648768           5835421440         0
          39318                  9202656
13858206 125000/125000           835682               6872648768           6864613920         0
          39318                  8026624

~~~

13873694 125000/125000           1936835682           15928536648768       15928522240320     0
          39318                  10033280
13873695 125000/125000           1936960682           15929564648768       15929551432800     0
          39318                  10041504
13873696 125000/125000           1937085682           15930592648768       15930580625280     0
          39318                  10008608
13873697 125000/125000           1937210682           15931620648768       15931609817760     0
          39318                  10008608
13873698 125000/125000           1937335682           15932648648768       15932639010240     0
          39318                  9613856
13873699 125000/125000           1937460682           15933676648768       15933668202720     0
          39318                  8446048
stream spooling ended at 13873700
=================== SUMMARY ====================
        Received Packets:1937460682
        Lost Packets:39318
        Buffered Stream Size: 15933676648768
        Saved File Name: wrtest_fila10g_2018y161d09h29m58s.vdif
        File Capacity: 15933676648768
================================================
```

그림 9. KREONET 상에서 8.274Gbps로 전송된 데이터 스트림을 16 x 1TB 디스크 어레이에 저장한 실험 결과
Fig. 9 Experimental result for the recording of 8.274Gbps data stream transferred over the KREONET to 16 X 1TB disk array

실험에서 링 버퍼를 경유하여 입출력된 프레임 개수는 1,937,460,682개이며 프레임 단위 용량은 8224바이트이다. 이를 위 식에 대입하면 기록된 데이터 총량은 15,933,676,648,768 바이트가 되며 이는 데이터 분석

결과가 출력된 하단의 SUMMARY와 완벽하게 일치한다. 이 실험 결과를 통해 TCP/IP 소켓 방식의 링 버퍼 시스템이 외부 네트워크와 내부의 가상 디스크 어레이 간 초고속 데이터 스트림 입출력을 효과적으로 수행하고 데이터 캡쳐와 저장에 있어서도 유용한 수단이 될 수 있음을 검증하였다.

## V. 결 론

본 논문에서는 데이터 소스에서 발생한 대용량 데이터를 TCP/IP 소켓 기반의 링 버퍼를 활용하여 원격의 스토리지에 저장하는 방안에 대해 기술하였다. 우리가 설계한 스토리지에서 네트워크를 통해 유입되는 관측 데이터 스트림은 TCP/IP 소켓 버퍼를 거쳐 시스템 상의 링 버퍼에 수신된 후 RAID 기반 스토리지에 순차적으로 저장된다. 이에 따라 네트워크 성능이 일정하게 유지된다면 입력 스트림에 대한 안정적인 수신은 물론 데이터 기록 및 분석 과정에 있어서도 KVN 각 사이트와 상관센터에 데이터를 중복 저장하는 비효율을 최소화시킬 수 있다. 데이터 분석에 있어서도 별도의 변환 작업 없이 리눅스 파일 형태로 즉시 상관 처리하는 것이 가능함에 따라 효율성과 생산성을 개선할 수 있다. 이를 입증하기 위해 KVN연세 사이트로부터 8Gbps 속도로 전송되는 관측 데이터를 원격의 상관센터에 바로 저장하였고, 데이터 분석을 통해 단 하나의 패킷 손실 없이 데이터가 완벽하게 저장되었음을 검증하였다. 본 논문에서는 UDP 기반으로 데이터 송수신 실험을 수행하였으나 향후 TCP로 확장할 경우 불안정한 네트워크 환경에서도 단 하나의 패킷 손실 없이 안정적인 데이터 기록이 가능할 것으로 예상된다.

## References


[1] J. Mcdonough, “Moving standards to 100 GbE and beyond,” *IEEE Communications Mag.*, vol. 45, no. 11, 2007, pp. 6-9.

[2] A. Whitney, C. Beaudoin, R. Cappallo, B. Corey, G. Crew, S. Doeleman, D. Lapsley, A. Hinton, S. McWhirter, and A. Niell, “Demonstration of a 16 Gbps Station-1 Broadband-RF VLBI System,” *Publications of the Astronomical Society of the Pacific,* vol. 125.924, no. 196, 2013. pp. 196-203.

[3] R. Cappallo, C. Ruszczyk, and A. Whitney, "Mark6: Design and Status," *Reports of the Finnish Geodetic Institute,* vol. 5, no. 8, Mar. 2013, pp. 9-12.

[4] A. Szomoru, "EXPReS and the e-EVN," *The 9th European VLBI Network Symp. on The role of VLBI in the Golden Age for Radio Astronomy and EVN Users Meeting,* vol. 72. Sept. 2009, pp. 1-8.

[5] S. Lee, L. Petrov, D. Byun, J. Kim, T. Jung, M. Song, C. Oh, D. Roh, D. Je, S. Wi, B. Sohn, S. Oh, K. Kim, J. Yeom, M. Chung, J. Kang, S. Han, J. Lee, B. Kim, H. Chung, H. Kim, H. Kim, Y. Kang, and S. Cho, "Early science with the Korean VLBI network: evaluation of system performance," *The Astronomical J.* vol. 147.4, no. 77, 2014, pp. 1-14.

[6] M. Song, H. Kim, D. Byun, T. Jung, Y. Kang, H. Kim, J. Kim, S. Wi, S. Lee, D. Roh, S. Oh, and J. Yeom, *The Design and Implementation of e-KVN Network for the Data Processing in Wideband VLBI Observation.* Daejoen: Korea Astronomy and Space Science Institute, 2019, pp. 3-5.

[7] D. Thompson and J. Best, "The future of magnetic data storage techology," *IBM J. of Research and Development,* vol. 44, no. 3, 2000, pp. 311-322.

[8] M. Song, Y. Kwang, and H. Kim, “Performance Evaluation of Big Stream based High Speed Data Storage,” *J. of the Korea Institute of Electronic Communication Sciences,* vol. 12, no. 5, Oct. 2017, pp. 817-827.

[9] J. Vetter and S. Mittal, “Opportunities for nonvolatile memory systems in extreme-scale high-performance computing,” *Computing in Science & Engineering,* vol. 17, no. 2, Jan. 2015, pp. 73-82.

[10] D. Luca, "Improving passive packet capture: Beyond device polling," In *Proc. of SANE,* Amsterdam, Netherlands, 2004, pp. 85-93.

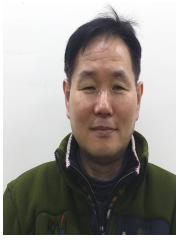
[11] S. Zhang, W. Li, Y. Zhang, and T. Wu, "Flow Control System Performance Optimization Based-On Zero-Copy," *International Conf. on Information Computing and Applications,* Chengde,

China, 2012, pp. 834-839.
[12] M. Song, Y. Kang, and H. Kim, "The Study on the Design and Optimization of Storage for the Recording of High Speed Astronomical Data," *J. of the Korea Institute of Electronic Communication Sciences,* vol. 12, no. 1, Jan. 2017, pp. 75-84.

## 저자 소개

**송민규(Gyu-Min Song)**

2001년 강원대학교 전기공학과 졸업(공학사)
2003년 강원대학교 대학원 전자공학과 졸업(공학석사)
2002년 ~현재 한국천문연구원 연구원
※ 관심분야 : 대용량 데이터 처리, 초고속 네트워크, 병렬 시스템



**김효령(Hyo-Ryoung Kim)**

1990년 서울대학교 천문학과 졸업(이학사)
1996년 부산대학교 대학원 천문학과 졸업(이학석사)
2003년 부산대학교 대학원 천문학과 졸업(이학박사)
1990년 ~현재 한국천문연구원 연구원
※ 관심분야 : 전파천문, 외부은하, 클러스터

**강용우(Yong-Woo Kang)**

1988년 부산대학교 기계설계학과 졸업(공학사)
1990년 부산대학교 대학원 지구과학과 졸업(이학석사)
2000년 부산대학교 대학원 지구과학과 졸업(이학박사)
2000년 ~ 2001년 연세대학교 박사후연구원
2002년 ~ 2006년 연세대학교 연구전임교원
2006년 ~현재 한국천문연구원 연구원
※ 관심분야 : 전파백엔드시스템, 대용량 자료처리, 관측천문학

**제도흥(Do-Heung Je)**

1992년 한양대학교 전자통신공학과 졸업(공학사)
1994년 한국과학기술원 전자전산학과 졸업(공학석사)
2002년 한국과학기술원 전자전산학과 졸업(공학박사)
2002년 ~ 현재 한국천문연구원 연구원
※ 관심분야 : 전파망원경 수신시스템 개발

**위석오(Seog-Oh Wi)**

1993년 전남대학교 전기공학과 공학사
1996년 전남대학교 전기공학과 공학석사
2002년 전남대학교 전기공학과 공학박사
2002년 ~현재 한국천문연구원 연구원
※ 관심분야 : KVN 시스템 유지보수 총괄, KVN전파망원경 유지보수, 서보시스템/Hexapod 시스템 개발

**이성모(Sung-Mo Lee)**

1998년 인천전문대학교 통신과 졸업
2010년 ~현재 한국천문연구원 연구원
※ 관심분야 : KVN전파망원경 유지보수, 전기전자 장비 관리