UDP 기반의 대용량 VLBI 데이터 전송 프로그램 개발†

(Development of UDP based Massive VLBI Data Transfer Program)

송 민 규*, 김 현 구***, 손 봉 원***, 위 석 오**,
강 용 우**, 염 재 환***, 변 도 영***, 한 석 태**
(Min-Gyu Song*, Hyun-Goo Kim***, Bong-Won Sohn***, Seog-Oh Wi**,
Yong-Woo Kang**, Jae-Hwan Yeom***, Do-Young Byun***,
and Seog-Tae Han**)

요 약 본 논문에서 우리는 대용량 데이터를 보다 효과적으로 전송할 수 있는 프로그램 구현 및 시스템 최적화에 대해 논의하고자 한다. 수백 ~ 수천 km 떨어진 전파망원경을 이용해 천체를 관측하는 VLBI에서 각 관측소는 수 TB에 달하는 대용량 데이터를 상관센터로 전송하여야 한다. 이에 따라 보다 효율적으로 데이터를 전송할 수 있도록 전세계적으로 초고속 네트워크를 활용한 e-VLBI 연구가 활발히 진행되고 있다. 본 논문에서는 이러한 연구 추세에 따라 초고속 네트워크 상에서 Gbps 급의 VLBI 데이터를 효과적으로 전달할 수 있는 프로그램 설계 및 시스템 구축에 대해 기술하였다. 기존의 VTP(VLBI Transport Protocol) 기반 데이터 전송을 넘어 보다 신속한 데이터 전달이 가능하도록 전송 프로그램 설계에 있어 UDP를 적용하였다. 이를 위해 현재 사용되는 VTP 프로그램에 Tsunami-UDP 알고리즘을 이식하여 데이터 전달에 있어 성능을 극대화하였고, 시스템 최적화를 통해 보다 신속하고 안정적으로 KVN 각 사이트 간 데이터 전송이 구현될 수 있도록 하였다.

핵심주제어 : e-VLBI, UDP, 대용량 데이터 전송, Tsunami-UDP, 초고속 네트워크

Abstract In this paper, we discuss the program implementation and system optimization for the effective transfer of huge amount of data. In VLBI which is observing the celestial bodies by using radio telescope hundreds thousands km apart, it is necessary for each VLBI observatory to transfer up to terabytes of observed data. For this reason, e-VLBI research based on advanced network is being actively carried out for the transfer of data efficiently. Following this trend, in this paper, we discuss design & implementation of system for the high speed Gbps data transfer rates. As a data transfer protocol, we use UDP for designing data transmission program with much higher speeds than currently available via VTP(VLBI Transport Protocol). Tsunami-UDP algorithms is applied to implementing data transfer program so that transmission performance could be maximize, also we make it possible to transfer observed data more fast and reliable through optimization of computer systems in each VLBI statopm.

Key Words : e-VLBI, UDP, massive data transfer, Tsunami-UDP, high speed network

† 이 논문은 2010년 한국천문연구원의 한국우주전파관측망운영 연구비 지원에 의해 연구되었음.
* 한국천문연구원 기술개발연구본부, 제1저자, 교신저자
** 한국천문연구원 기술개발연구본부, 공동저자
*** 한국천문연구원 전파천문연구본부, 공동저자

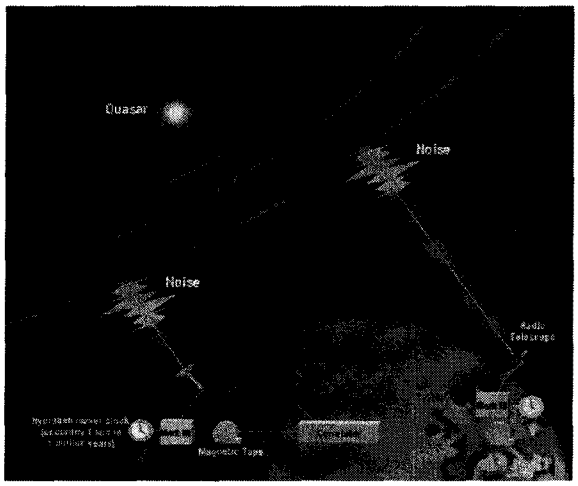
1. 서 론

네트워크 기술의 급격한 발전 및 진보에 따라 다양한 연구 분야에서 e-Science가 적용되어 활용되고 있다. 고에너지 물리, 기상은 물론 의학, 지질, 생명공학 연구 등에 이르기까지 네트워크 기술을 접목한 e-Science는 다수의 연구 분야로 파생되고 있으며 실제 그에 걸맞는 연구 성과를 내고 있다[1]. 상기 분야와 마찬가지로 e-Science가 폭넓게 활용되고 활성화되어 있는 분야로 천문 분야를 들 수 있으며, 그 중 VLBI의 경우 다수의 전파망원경으로부터 얻은 데이터를 상관센터에서 처리하는 특성으로 인해 안정적인 초고속 네트워크 확보가 반드시 전제되어야 한다. VLBI 연구와 관련하여 한국천문연구원에서는 KVN (Korean VLBI Network)을 일차적으로 구축 완료하여 현재 운용 중에 있으며, 관측소에서 얻은 수 TB의 데이터를 보다 효과적으로 상관센터로 전송할 수 있도록 세 관측소와 상관센터 간에 초고속 네트워크를 구축하였다. 데이터 전달에 있어 물리적 네트워크와 더불어 중요한 요소로서 최적화된 프로토콜 개발 및 선택을 들 수 있다. 본 논문에서는 Gbps급의 대용량 데이터 전송을 위한 기반 프로토콜로 Tsunami-UDP를 지정하였고 그를 기존의 VTP(VLBI Transport Protocol)에 접목시킴으로써, 데이터 전송의 효율을 극대화하고자 하였다[2]. Tsunami-UDP 기술을 적용하여 개발된 프로그램과 기존의 VTP 기반의 프로그램에 대해 실제 데이터 전송을 시연함으로써 각 방식의 성능 분석은 물론, KVN의 데이터 전달에 있어 최상의 해법을 찾을 수 있도록 하였다. 본 논문은 다음의 순서에 따라 구성되었다. 서론에 이어 2장에서는 VLBI 및 e-VLBI에 대한 소개 및 간략한 원리에 대해 설명할 것이고 3장에서 e-VLBI 구현을 위한 e-KVN에 대해 시스템 구성을 기반으로 설명하고자 한다. 4장에서 KVN 각 사이트 간 데이터 전송을 위한 기존의 VTP 방식 및 프로그램에 대해 살펴보는 것은 물론 그를 개선하기 위한 UDP 기반의 Tsunami-UDP에 대해 알아보기로 한다. 5장에서는 Tsunami-UDP 알고리즘을 적용하여 네트워크 상에서 보다 효율적으로 데이터를 전송할 수 있는 프로그램을 설계 및 구현할 것이고 6장에서 그에 대한 데이터 전송 및 성능 분석을 진행하고자 한다. 그리고 마지막으로 7장에서 본 논문의 결론을 맺기로 한다.

2. e-VLBI 개요 및 특성

VLBI는 지난 30년 이상 전파천문을 비롯한 지구물리, 측지 등의 여러 학문 분야에서 널리 사용되어 온 기술로서 천체 관측에 있어서 고분해능의 공간 정밀도를 제공하며 지구 자전축 및 지각 운동의 미세한 변화를 측정함에 있어서 가장 정확한 방법 중의 하나이다[3]. 본 장에서는 이러한 특성을 갖는 VLBI 및 네트워크 기반의 e-VLBI에 대해 살펴본다.

2.1 VLBI 소개



<그림 1> VLBI 시스템의 기본 메커니즘

각 전파망원경에서 천체로부터 수신하여 얻은 전파 신호를 서로 합성하면, 그 위상(phase)의 차이에 따라 간섭이 일어나 각 전파망원경 간의 거리에 비례하는 직경의 초대형 전파망원경으로 천체를 관측한 것과 같은 효과를 얻을 수 있는데 이러한 시스템을 전파간섭계라 한다[3]. 전파간섭계는 전파망원경에서 나오는 전파 신호를 합성하는 방법에 따라 크게 두 가지 형태로 분류되는데 그 중의 한 형태로서 전파망원경을 케이블로 직접 연결하여 전파 신호를 합성하는 전파망원경 어레이(array)가 있다. 한편 전파망원경

어레이 보다 더욱 높은 공간 분해능을 얻기 위해서는 상호 거리가 수 백 또는 수 천 km 떨어져 있는 전파망원경을 사용해야 하는데, 이러한 경우 케이블을 이용하여 여러 전파망원경을 직접 연결하는 것이 불가능하기 때문에 각 전파망원경으로부터 얻은 신호를 자기 테이프에 기록한 다음 이 테이프들을 한 곳에 모아서 자료를 합성하는 절차를 거치게 된다. 이러한 장치를 VLBI(Very-Long- Baseline Interferometry:초장기선 전파간섭계)라 부르며, 이를 통하여 기존의 단일 전파망원경이나 전파망원경 어레이 보다 훨씬 높은 공간 분해능의 천체 이미지를 얻는 것이 가능하다. 위 <그림 1>에서는 두 전파망원경으로 구성되는 VLBI 어레이의 동작 메커니즘을 간략히 나타내었다.

VLBI에서 각 전파망원경은 서로 독립되어 운용되기 때문에 전파망원경 간의 거리는 원하는 대로 확장 가능하며 이로부터 분해능의 이점을 누릴 수 있지만 다음 두 가지 점에서 신중을 기해야 한다. 첫째는, 표준 주파수의 정확한 유지이고 둘째는, 대기에 의해 발생하는 위상 요동을 최소화하는 적절한 위상 보정이다. 이 외의 부분은 서로 연결되어있는 전파망원경 어레이와 원리적으로 유사하다. 표준 주파수의 미세한 차이는 위치 및 지구 운동 등에 엄청난 오차를 가져오고, 위상 보정이 잘 안된 데이터는 천체로부터 온 신호의 간섭성을 떨어뜨리기 때문에 자료의 질적 악화를 가져온다. 그러므로 각 관측소에서 안정된 표준 주파수 유지와 효율적인 위상 보정 방법을 확보하는 것이 VLBI 시스템 운용에 있어 가장 중요하다 할 수 있다[3].

2.2 네트워크 기반 e-VLBI의 필요 및 효율성

기존 방식의 VLBI에서는 각 관측소에서 얻은 데이터를 자기 테이프나 디스크에 기록한 다음 배나 비행기 등의 운송 수단을 통하여 데이터센터로 전송한 후 영상 합성·처리 하였다. 하지만 이 방식은 데이터 기록이나 테이프 운송에 있어서 많은 비효율 및 고비용의 문제를 야기시켰고 무엇보다 관측 결과에 대한 신속한 분석이 불가능하다는 점에서 크나큰 취약점을 갖고 있었다. 이러한 문제점을 극복하기 위하여 등장한 기술이 e-VLBI로서 이를 통하여 Gbps급의 VLBI 데이터를 실시간 또는 준실시간으로 전송하는 것이 가능하기 때문에 데이터 처리에 있어서의 신속성, 분석의 정확성은 물론이고 시스템 운용에 있어서의 경제적 효과 또한 얻을 수 있다. 각 관측소에서 얻어진 VLBI 데이터는 전송 가능한 형태로 변환되어 초고속 네트워크를 통해 데이터센터로 전송되는데 이러한 e-VLBI의 활용을 통해 얻을 수 있는 장점을 간략히 요약해보면 다음과 같다.

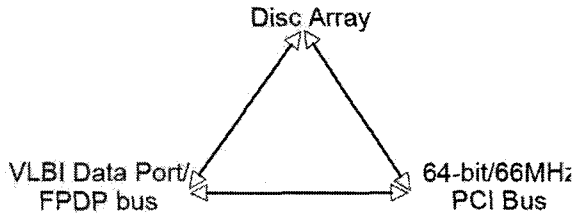
- 신속한 데이터 처리 및 분석의 정확성
- 보다 정밀한 관측 가능
- 경제적 시스템 운용
- 간편하고 효율적인 시스템 진단

3. e-KVN의 네트워크 및 시스템 구성

e-KVN은 네트워크 기반의 VLBI 시스템으로서 KVN의 세 관측소 및 상관센터를 초고속 네트워크로 연결하여 실시간 또는 준실시간 VLBI 관측을 수행하는 것을 목표로 한다[4]. 본 논문에서는 대용량 데이터 전송이 구현되는 e-KVN의 VLBI 시스템 및 네트워크 구성에 대해 알아보기로 한다.

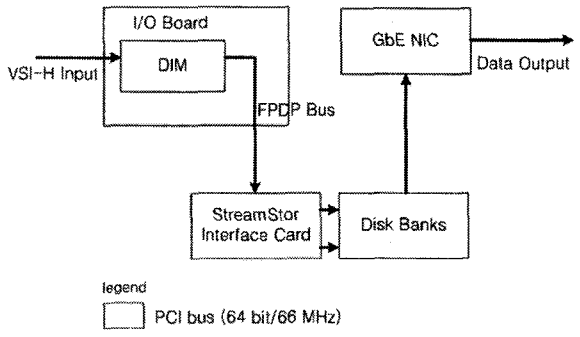
3.1 VLBI 데이터 전송 시스템

전파망원경이 설치된 각 관측소에서 얻어진 아날로그 형태의 천체신호는 수신기, 스펙트로미터, 디지털필터뱅크 등을 거쳐 Mark5B라 불리는 데이터 저장시스템에 Gbps의 속도로 기록된다. Mark5B는 미국 MIT의 헤이스택 관측소에서 개발된 대용량 데이터 기록/재생 시스템으로서 전세계의 전파 관측소에서 널리 활용되고 있다. KVN의 경우도 관측소와 상관센터에서 각각 데이터에 대한 초고속 저장, 재생 처리용으로 사용하고 있으며 이에 대한 핵심적 인터페이스에 해당하는 StreamStor 카드를 중심으로 Mark5B의 기능 및 시스템 구성을 간략히 살펴보면 아래와 같다[2].



<그림 2> Mark5B의 물리적 인터페이스

Mark5B의 메커니즘은 위의 세 인터페이스 간의 데이터 송수신에 기반한다. 세 물리적 인터페이스 중 임의의 두 인터페이스 사이에서 데이터는 어느 방향으로든 이동가능하며 최대 1200Mbps의 데이터 전송속도를 지원한다. 한번에 오직 하나의 연결 경로만이 사용 가능하고 실제적으로 VLBI 실험에서 활용되는 최대 데이터 전송속도는 1024Mbps이다. 기존 형태의 VLBI는 Mark 5B에서 FPDP 버스와 디스크 어레이 간의 경로를 통하여 데이터가 전달되는 방식에 해당한다. FPDP 버스와 PCI 버스간의 데이터 흐름의 경우 데이터는 디스크에 기록되지 않는다. 따라서 향후 DAS로부터 유입되는 데이터를 직접 초고속 네트워크로 전송할 수 있는 완벽한 실시간 데이터 전송을 구현할 수 있다. 본 논문에서는 이러한 완벽한 실시간 데이터 전송 이전에 Mark5B의 디스크에 기록된 대용량 데이터를 네트워크를 통해 효율적으로 전송하는 시스템 아키텍처를 구현하고자 하며 그에 대한 동작 메커니즘을 다음과 같이 나타낼 수 있다.



<그림 3> 관측소에서의 데이터 전송 메커니즘

VLBI 데이터를 기록 및 재생하기 위해서는 포맷터, 상관기, VSI 등 일반 형태의 VLBI 데이터 인터페이스와 32 비트 FPDP 버스 사이에 변환이 필요하며 Mark5B에서 I/O보드는 소프트웨어 설정에 따라 데이터 입력 모듈(DIM: Data Input Module)과 데이터 출력 모듈(DOM: Data Output Module)로 동작된다. VSI-H 인터페이스를 통하여 입력된 데이터는 FPDP 인터페이스 지원의 I/O 보드를 통해 StreamStor 카드를 경유해 디스크에 저장된다[2]. 이렇게 디스크에 기록된 관측 데이터는 DOM 기능의 I/O 보드를 통해 VSI-H 포맷으로 전송된다. 하지만 이는 VSI 케이블로 연결된 인접한 시스템으로 데이터를 전송하는 경우에 해당되며 서로 멀리 떨어진 시스템의 경우에는 네트워크를 통한 데이터 전달이 구현되어야 한다. 이를 구현함에 있어 필요한 것으로 초고속 네트워크와 데이터를 효과적으로 전송할 수 있는 애플리케이션을 들 수 있으며, 다음 절에서 그를 위한 초고속 네트워크에 대해 기술하고자 한다.

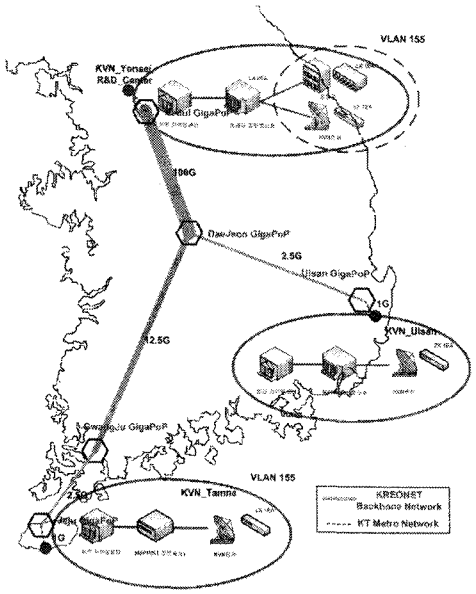
3.2 네트워크 아키텍쳐

e-KVN은 KVN을 구성하는 세 전파천문대(연세대, 울산대, 탐라대) 및 상관센터를 Gbps 급의 초고속 네트워크로 연결하여 대용량 데이터를 실시간으로 전달하기 위한 네트워크 기반의 VLBI 시스템이라 할 수 있다. 대용량의 관측 데이터를 실시간 또는 준-실시간으로 상관센터로 전송하는 것이 가능하기 위해서는 그를 지원하는 초고속 네트워크가 구비되어야 한다[5]. 이를 위하여 KVN의 각 관측소와 상관센터에 KREONET 기반의 네트워크를 구성하였고, 그를 통해 사설망 수준의 데이터 전송이 가능하도록 하였다[6].

위 그림은 이를 나타내는 e-KVN의 개략적 네트워크 구성도로서 적절한 프로토콜 설계 및 프로그램 활용 시 Gbps 수준의 데이터 전송이 구현가능하리라 생각한다. 하단의 물리적 시스템과 더불어 다음 장에서는 보다 효율적인 데이터 전송을 위한 프로토콜 설계 및 활용, 개발에 대해 기술하기로 한다.

4. VLBI 데이터 전송 프로토콜

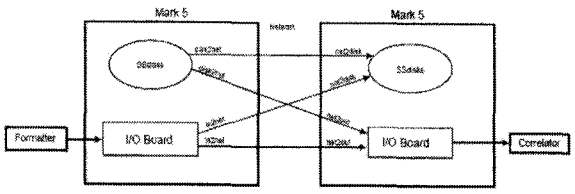
이전 장에서 VLBI 데이터 저장/재생 시스템인



<그림 4> 데이터 전송을 위한 e-KVN 백본망 아키텍쳐

Mark5B에 대해 살펴보았는데, 본 절에서는 현재 해당 시스템에서 데이터를 전송하는 프로토콜에 해당하는 VTP(VLBI Transport Protocol)의 기능 및 메커니즘에 대해 먼저 살펴보기로 한다. 이후 데이터 전송 성능 개선을 위하여 UDP 기반의 Tsunami-UDP에 대해 알아보는 것은 물론, 그를 기존의 프로그램에 적용할 방안에 대해 모색하고자 한다.

4.1 VTP를 이용한 데이터 전송



<그림 5> 네트워크를 경유한 Mark 5 시스템 간의 데이터 전송

VTP는 VLBI Transfer Protocol의 약자로서 네트워크 상에서 VLBI 데이터 전달을 위해 미국 MIT의 헤이스택 관측소에서 개발된 라이브러리이다. 각 관측소와 상관센터에 설치된 Mark5B 시스템 간에 소켓 인터페이스를 기반으로 데이터를 전송하며 커맨드 라인 인터페이스를 기준으로 동작한다[2]. 데이터 전송을 위

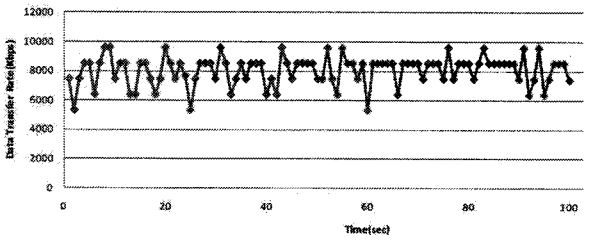
<표 1> VLBI 데이터 전송을 위한 VTP 라이브러리

	기능
File2net	Mark5 시스템의 파일을 네트워크로 전송
Net2file	전송된 데이터를 파일 형태로 수신
Disk2net	Mark5 시스템의 스캔을 네트워크로 전송
Net2disk	전송된 데이터를 스캔 형태로 수신

한 프로토콜은 전송 및 수신하고자 하는 데이터의 형태에 따라 크게 두 가지 형태로 구분된다. 컴퓨터 시스템의 일반 데이터를 기준으로 데이터 전송을 수행하는 경우에는 송신단과 수신단에서 각각 File2net, Net2file이 사용된다. 반면 디스크 뱅크에 저장된 스캔 데이터에 대한 송수신에는 Disk2net, Net2disk가 호출되며, 이러한 네트워크 상의 데이터 전송을 통해 교통수단을 통한 디스크 모듈 운송 과정에서 발생하는 시간지연 및 비용발생 문제를 극복할 수 있다. 상기 살펴본 VTP 기반 데이터 전송 프로토콜의 기능 및 데이터 송수신 메커니즘을 그림으로 요약해보면 위 <그림 5>, <표 1>과 같다.

```
## 관측소에서의 데이터 송신
void * toX(void * null) {
    /* * Process output data transfers for in2net and disk2net (i.e.,
     * write to net), net2out (i.e., write to output board), and
     * net2disk (write to SS disk).  See also Xto(). * */
...
  if (ot->datah == 4) /* Disk2net? */
    scbuf = MIN(socbuf, ot->endbyte - ot->nowbyte); /* Yes, size to send
*/
  else /* In2net, datah = 5 */
    scbuf = socbuf; /* Always send a whole segment */
  len = send(ports[0], rbuf + index * indst, scbuf, 0);
      /* (Remember:  long long rbuf[]) */
  if (len != scbuf) { /* OK? */
## 상관센터에서의 데이터 수신
void * Xto(void * null) {
    /* * Process input data transfers for in2net (i.e. read from input
     * board), disk2net (i.e., read from disk), net2out, and net2disk
     * (i.e., read from net).  See also toX(). * */
...
  if (first && ot->msglev < 1) /* First debuggery? */
    (void) fprintf(stderr, /* Yes */
        "%s Xto() DEBUG:  Doing recv(), index %d \n", me, index);
  lenn = recv(ports[0], rbuf + index * indst, socbuf, MSG_WAITALL);
      /* (Remember:  long long rbuf[]) */
      /* * Various other checks * */
  if ((unsigned) lenn < socbuf && ot->msglev < 1) /* Short and
debuggery? */
```

<그림 6> VLBI데이터 송수신을 위한 기존 프로그램의 소스코드



<그림 7> VTP 활용하의 데이터 전송 성능

하지만 VTP를 이용한 데이터 전송의 경우 가장 큰 단점으로 수준 이하의 데이터 전송 성능을 들 수 있다. VTP의 경우 TCP/IP 소켓을 기반으로 구동되며 각 관측소와 상관센터의 시스템에서 단순히 send(), recv() 함수를 활용하여 데이터 송수신 메커니즘을 구현하는데 이에 대한 소스코드 및 데이터 전송 성능을 그래프로 나타내면 아래와 같다[2].

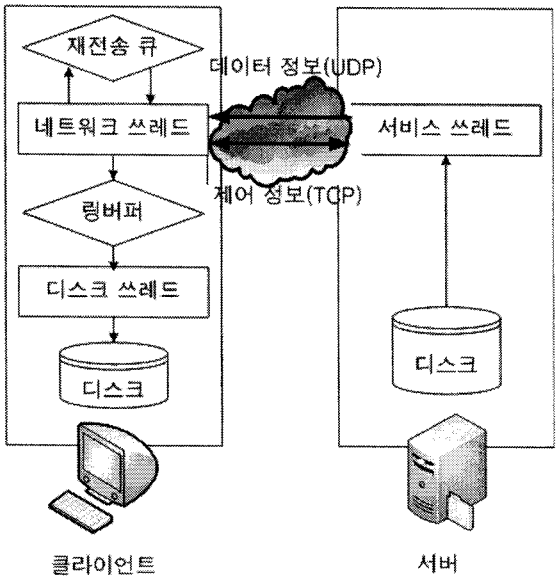
기존의 VTP 기반 애플리케이션의 성능을 시험하기 위해 100MB 크기의 데이터를 송신단에서 네트워크를 통해 수신단으로 전송하였다. 전체 파일을 전송에 100초 남짓한 시간이 소요되었고 평균 전송속도는 8025.99blocks/sec(64.2Mbps)가 산출되었다. 고성능의 네트워크 사양에 비춰보았을 때 이는 대단히 낮은 전송 속도에 해당하며 명백한 병목현상을 나타낸다. 통상적으로 송수신 시스템에 크나큰 문제가 있다는 것을 나타낸다. 네트워크 대역폭이 1Gpbs라는 것을 감안하였을 때 통상적으로 네트워크 자체의 문제라면 전송 속도가 제공 대역폭의 1/10에는 근접한 수치가 나와야 한다. 하지만 데이터 전송 속도가 애초 할당받은 대역폭의 1/100에도 미치지 못하는 것은 근원적인 문제가 전송 시스템에 있음을 의미하며 하드웨어 자체보다는 전송 소프트웨어에 보다 큰 결함이 있음을 나타낸다. 이를 극복하기 위해서는 네트워크 상의 데이터 전송에 최적화된 프로토콜 적용 및 애플리케이션 내에서의 구현이 필요하며 이에 따라 본 논문에서는 UDP 기반의 Tsunami-UDP를 데이터 전송에 활용함으로써 초고속 네트워크의 장점을 효과적으로 이끌어내고자 한다.

4.2 Tsunami-UDP 기반의 데이터 전송

TCP가 개발되던 초창기에는 네트워크 상에서 트래픽 혼잡으로 인한 패킷 손실은 TCP 메커니즘에 의해 상당부분 치유가 가능하였다. 즉, TCP의 흐름제어 메커니즘에 의해 제한된 네트워크 자원을 효율적으로 사용하는 것이 가능하였다. 하지만 현 네트워크는 높은 QoS를 바탕으로 네트워크 혼잡이 최소화되는 형태로 발전하고 있다. 더욱이 수 Gbps 대역폭 전용 할당의 연구망이 급속도로 팽창하는 시대에 TCP는 대용량 데이터 전달에 있어 여러 문제점을 가지고 있다. 그 중 네트워크에서 발생하는 패킷 손실을 혼잡상황 발생으로 간주하는 것은 성능 저하의 가장 큰 요인으로서 네트워크의 대역폭이 아무리 방대하여도 데이터 전달 성능을 떨어뜨리는 결과를 초래한다. 데이터 유실이 한번이라도 발생하면 TCP의 혼잡 회피(congestion avoidance) 알고리즘이 구동하여 전송 윈도우의 크기가 즉각적으로 감소되고 이로 인해 활용 가능한 대역폭도 실제적으로 감소하였다[7]. 반면 가용 네트워크가 늘어나기 위해서는 데이터 유실이 발생하지 않으면서 상당한 시간을 기다려야 했는데 바로 이러한 특성으로 인해 TCP 기반의 데이터 전송에 있어 만족할만한 성능을 기대하는 것이 불가능하였다. Tsunami-UDP 프로토콜은 UDP 기반의 프로토콜로서 높은 BDP(Bandwidth Delay Product)를 갖는 네트워크 경로상에서 초고속으로 파일을 전송하기 위해 개발되었다. 기존의 표준 TCP 프로토콜로는 높은 대역폭, 긴 RTT 특성의 초고속 네트워크 상에서 만족할만한 데이터 전송을 기대할 수 없었다. 이러한 한계를 극복하기 위해 Tsunami-UDP, Fast TCP 등과 같은 대용량 데이터 전송에 최적화된 프로토콜이 개발되기에 이르렀다. Tsunami-UDP는 전송하고자 하는 파일을 32kB 크기의 블록으로 분할하여 데이터를 전송에 있어 효율성을 극대화하였다[8]. Tsunami-UDP는 기존 TCP의 슬라이딩 윈도우 메커니즘이 아닌 패킷 간 지연 시간 조정을 통한 데이터 전송 제어를 특징으로 한다. 뿐만 아니라 데이터를 송수신하는 클라이언트와 서버 간의 세션 유지에 필요한 통신은 TCP 연결을 통해 수행되며 대용량 데이터 전달은 UDP 프로토콜을 기반으로 구현되는데 데이터 전송에 있어 동작 메커니즘을 도시해보면 다음과 같다.

파일 전송 과정에서 클라이언트는 네트워크와 디스크를 담당하는 두 개의 쓰레드를 실행한다. 네트워크 쓰레드는 모든 네트워크 통신을 처리하고 재전송 큐

를 관리할 뿐만 아니라 디스크 스토리지에 쓰일 데이



<그림 8> Tsunami-UDP의 구성 및 동작 메커니즘

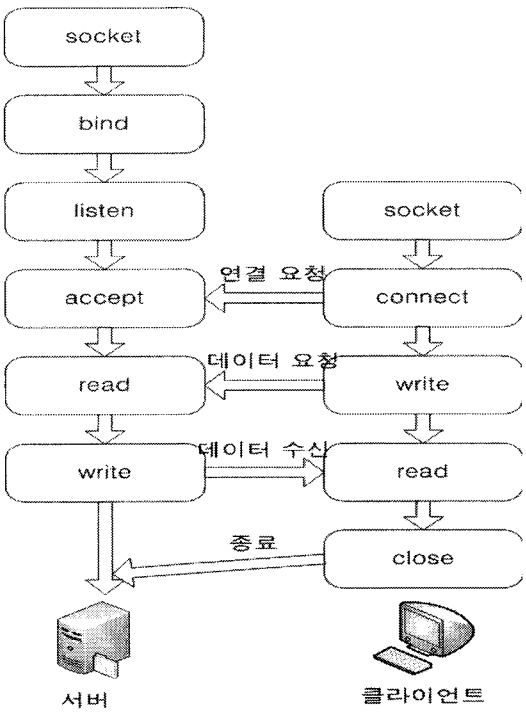
터 블록을 링 버퍼에 보관한다. 디스크 쓰레드는 데이터 블록을 링 버퍼로부터 디스크의 파일로 이동시킨다. 서버는 각 클라이언트의 연결 요청에 대응하여 디스크와 네트워크 역할을 수행하는 단일 쓰레드를 생성한다. 서버로부터 데이터를 전송 받기 위해 클라이언트는 파일 이름을 서버측으로 전달한다. 이후 원하는 블록 사이즈, 전송 속도, 에러 한계치, 패킷 간 지연 팩터를 서버 측으로 함께 전달하며 서버는 이에 대한 응답으로 파일 크기, 합의된 블록 사이즈, 블록 개수, 타임스탬프를 클라이언트 측으로 전송한다. 데이터 전송을 위해 클라이언트와 서버 간의 UDP 세션이 성립되면 서버는 랜덤 형태의 데이터의 작은 블록을 클라이언트로 전송한다.

5. 대용량 데이터 전송 프록그램 설계 및 구현

기존의 TCP 기반으로는 데이터 전송에 구조적 한계에 부딪힐 수 밖에 없다는 것에 대해 확인하였고 그를 개선하기 위한 방안으로 UDP기반의 Tsunami-UDP 프로토콜에 대해 알아보았다. 본 절에서는 네트워크 상에서 대용량 데이터 전송을 위한 프로그램 설계 및 구현 방안에 대해 논의하고자 한다.

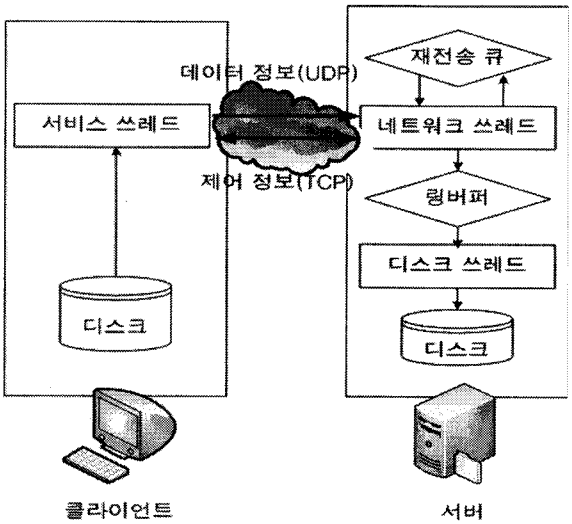
5.1 프로그램 설계

VLBI는 수 백 ~ 수 천km 떨어진 전파망원경에서 동일한 천체를 관측하여 데이터를 얻으면 상관센터에서 그에 대한 처리 및 분석을 하는 메커니즘으로 진행된다. 전파망원경이 위치한 각 관측소에는 수 TB에 이르는 막대한 양의 데이터가 발생한다. 이를 네트워크를 통해 상관센터로 효과적으로 전송하는 것이 e-VLBI로서 그를 구현함에 있어 초고속 네트워크와 업그레이드된 프로토콜을 전제로 한다. 각 관측소에서 획득한 데이터를 상관센터로 전송한다는 점에서 각 VLBI 사이트는 클라이언트와 서버에 대비된다. 초고속 네트워크 상에서 데이터를 송신하는 관측소는 상관센터에 대한 연결 요청 및 세션을 수립한다. 반면 상관센터는 관측소로부터의 데이터 전송을 기다리며 대기하다가 요청이 들어오면 그에 대한 수락 및 데이터를 수신받는 서버로서의 역할을 수행한다. TCP/IP 네트워크의 관점에서 각 관측소와 상관센터의 이러한 메커니즘을 <그림 9>에 나타내었다.



<그림 9> 데이터 전송을 위한 관측소-상관센터간의 시스템 및 정보 교환

관측소와 상관센터의 시스템은 소켓 개설 및 연결 요청 및 수락 절차를 통해 TCP/IP 소켓 인터페이스 기반으로 세션을 생성한다. 관측소와 상관센터 간의 연결 설정 및 제어 정보 전달을 위한 세션이 생성되며 이는 TCP 스트림 소켓으로 구현된다. 이전 장에서 언급하였듯이 대용량 데이터 전송에 있어 TCP의 혼잡 제어, 슬라이딩 윈도우는 성능 감소 요인으로 작용하며 높은 대역폭, 긴 지연시간을 갖는 초고속 네트워크에서 이러한 상황은 더욱 가중된다. 데이터 전송에 있어 제어 및 네트워크 상태 등의 정보는 TCP로 처리하되 실제적 데이터 전달은 UDP 기반으로 구현할 필요가 있으며 이에 따라 본 논문에서는 Tsunami-UDP를 활용하여 대용량 데이터 전달이 가능하도록 하였다. Tsunami-UDP를 KVN 관측소와 상관센터 간의 데이터 전송에 활용함에 있어 개선되어야 할 부분이 있다. 이는 Tsunami-UDP와 KVN 사이트 간의 데이터 전달 방식과 연관된다고 할 수 있는데 Tsunami-UDP는 기본적으로 클라이언트가 서버에 접속한 후 데이터를 내려받는 다운로드 방식을 취한다. 하지만 VLBI에서 데이터 전송 방식은 관측소에서 상관센터로 전달하는 형태로서 클라이언트에서 서버로 데이터를 업로드하는 방식에 해당한다. 데이터 전송에 있어 각 관측소와 상관센터 간의 이러한 관계는 이미 <그림 1>을 통해서도 확인되었다. 이는 기존 Tsunami-UDP



<그림 10> VLBI 데이터 전송을 위한 개선된 형태의 Tsunami-UDP

에서 보여진 다운로드를 통한 데이터 획득과는 정반대의 형태로서 Tsunami-UDP 알고리즘을 e-VLBI에 효과적으로 적용하기 위해서는 Tsunami-UDP의 구성을 다음과 같은 형태로 변경할 필요가 있다.

각 관측소와 상관센터 간의 데이터 전송은 커맨드 라인 인터페이스에 기반한다. 상관센터에 위치한 서버에서 데몬의 형태로 데이터 수신 프로그램이 실행되면 관측소의 클라이언트에서 접속 후 데이터 전송이 가능하며, 네트워크 연결을 위해 서버 시스템의 IP 주소, 포트 넘버, 전달하고자 하는 파일이름이 필요하나, 통상적으로 Tsunami-UDP의 경우 포트 넘버로서 46224를 사용하기에 커맨드 라인에서 별도로 지정할 필요는 없도록 하였다. VLBI에서 관측소와 상관센터 간 데이터 전송을 위해 실행되어야 할 커맨드 아키텍처는 <그림 11> 같이 나타낼 수 있다. 관측소의 클라

```
클라이언트
        client <IP_Address> <전송 파일>
서버
        server
```

<그림 11> 데이터 전송을 위한 각 시스템 상의 커맨드 아키텍처

이언트 시스템 상에서 사용자는 현 디렉토리 내에 국한되지 않고 임의의 파일 시스템, 디렉토리에 위치하는 파일을 지정하여 전송 가능하도록 하였다. 상관센터로 전달된 데이터는 클라이언트에서와 동일한 파일 이름으로 서버 시스템에 저장되며 파일 전송 완료 시 클라이언트의 소켓은 차단되어 서버와의 세션은 종료되도록 하였다. 단 서버의 경우 데이터를 지속적으로 수신하는 상관센터의 특성을 고려하여 데몬 형태로 데이터 수신 프로그램이 실행되도록 설계하였다.

데이터 전송에 있어서 효율성 배가를 위해 본 논문에서는 UDP 기반의 Tsunami-UDP 알고리즘을 적극 활용하였다. 작은 패킷 손실도 혼잡 상황으로 규정하여 윈도우 사이즈를 급감시키는 기존 프로토콜의 단점을 지양한 대신 허용 가능한 패킷 유실의 한계치를 기준으로 패킷 간 지연 시간을 적절하게 늘리거나 줄이는 방법으로 데이터 전송에 있어 네트워크 성능을 최대로 이끌어낼 수 있도록 하였다. 이와 더불어 클라

이언트 사용자가 데이터 전송 관련 원하는 블록 사이즈, 전송 속도, 에러 한계치, 패킷 간 지연 팩터를 설정하도록 하는 Tsunami-UDP의 핵심 알고리즘을 그대로 차용함으로써 대용량 데이터 전송에 있어 최상의 성능이 구현될 수 있도록 하였다.

5.2 프로그램 구현

VLBI 관측소에서 획득한 데이터를 상관센터로 전송하기 위해서는 일차적으로 클라이언트에서 서버 측으로 접속 요청 및 제어 세션 성립이 이뤄져야 한다. 제어 세션은 클라이언트, 서버의 시스템 정보, 네트워크 주소, 송수신 파일이름 교환을 위한 세션으로서 데이터 전송을 위한 클라이언트의 접속 요청 처리를 위해 서버 상에서는 관련 프로세스가 실행되어야 한다. 그를 위한 소켓 개설 및 세션 연결을 각 시스템 별로 다음과 같이 구현하였다.

```
int main(int argc, const char *argv[])
{
        ...
        /* create a server socket */
        if((server_fd=socket(PF_INET,SOCK_STREAM,0))<0){
        ...
        }
        /* setting of socket option */
        if((status=setsockopt(server_fd,SOL_SOCKET,SO_REUSEADDR,&yes,\
        sizeof(yes)))<0){
        ...
        }
       /* initializing of socket structure to '0' */
        memset(&server_addr,0,sizeof(server_addr));
       /* setting of server_addr */
        server_addr.sin_family=AF_INET;
        server_addr.sin_addr.s_addr=htonl(INADDR_ANY);
        server_addr.sin_port=htons(DEFAULT_SERVER_PORT);
       /* bind function call */
        if((bind(server_fd,(struct sockaddr*)&server_addr,sizeof(server_addr)))
        <0){
        ...
        }
        printf("Server started.\nWaiting for client..\n");
        listen(server_fd,5);
        clilen=sizeof(client_addr);
        while(1){
       /* waiting for connection request from client */
                if((client_fd=accept(server_fd,(struct
sockaddr*)&client_addr,&clilen))<0){
        ...
        }
```

```
int main(int argc, char *argv[])
{
        ...
        /* command and argument needed to transfer data */
        if (argc!=3){
                printf("Usage: %s <Server_IP> <File_Name>\n",argv[0]);
                exit(0);
        }
        /* file name handling */
        filename_ptr=strrchr(argv[2],'/');
        if (filename_ptr==NULL)
                strcpy(filename,argv[2]);
        else
                strcpy(filename,filename_ptr+1);
        session.transfer.filename=filename;

        /* create a socket */
        if((server_fd=socket(PF_INET,SOCK_STREAM,0))<0){
        ...
        }
        /* setting of socket option */
        if((status=setsockopt(server_fd,SOL_SOCKET,SO_REUSEADDR,&yes,
        sizeof(yes)))<0){
        ...
        }
        if((status=setsockopt(server_fd,IPPROTO_TCP,TCP_NODELAY,&yes,
        sizeof(yes)))<0){
        ...
        }
        /* initializing of socket structure to '0' */
        memset(&serv_addr,0,sizeof(serv_addr));
       /* setting of server_addr */
        serv_addr.sin_family=AF_INET;
        serv_addr.sin_addr.s_addr=inet_addr(argv[1]);
        serv_addr.sin_port=htons(DEFAULT_TCP_PORT);
        /* request of connecting to server */
        if((status=connect(server_fd,(struct sockaddr*)&serv_addr,sizeof\
        (serv_addr)))<0){
        ...
        }
        printf("Client is connected to remote server successfully!!\n");
        ...
}
```

<그림 12> 데이터 전송을 위한 접속 요청 및 네트워크 세션 생성

위 과정을 통해 VLBI 관측소 내의 클라이언트 시스템과 상관센터 서버 간의 네트워크 세션을 수립하였고 데이터 전송을 위한 아키텍쳐를 구현하였다. 이러한 네트워크 세션 상에 보다 신속하고 효과적으로 데이터를 전송할 수 있는 메커니즘을 구현하기 위해 본 논문에서는 Tsunami-UDP 알고리즘을 적용하였다[8]. 클라이언트는 <그림 11>에 소개한 명령 중 세 번째 인자를 통해 전달하고자 하는 파일 이름을 지정하며, 네트워크 상의 데이터 전송을 위해 사전에 오픈되어야 한다. 보다 효율적인 데이터 전송이 이뤄질 수 있도록 클라이언트는 블록 크기, 목표로 하는 데이터 전송 속도, 오차 허용율 등의 정보를 목록화하고 파일의 크기, 블록 사이즈, 블록 수 등의 정보를 기반으로 마지막 블록에 도달할 때까지 루프를 돌며 데이터를 전송할 수 있도록 하였다. 본 논문에서는 데이터 전송에 있어 TCP가 아닌 비연결형 데이터그램 전송

방식의 UDP 방식에 기반을 두었다. 신속하고 효과적인 데이터 전달은 가능하나 흐름제어를 하지 않는 UDP의 특성상 데이터 유실 가능성은 증가할 수 밖에 없다. 이러한 단점을 보완하기 위하여 TCP의 특성에 해당하는 재전송 기능을 별도의 구조체를 빌어 구현하였는데, 이상 살펴본 메커니즘 구현을 위한 클라이언트의 의사코드를 나타내면 아래와 같다.

```
    /* designation of filename */
        session.transfer.filename=filename;
    /* try to open the file for reading */
    xfer->file = fopen(xfer->filename, "r");
    if (xfer->file == NULL) {
    ...
    }
    /* reply with the length, block size, number of blocks, and run epoch */
    file_size    = htonll(param->file_size);      if (full_write(session->client_fd,
&file_size,   8) < 0) return warn("Could not submit file size");
    ...
    /* open a new datagram socket */
    session->transfer.udp_fd = create_udp_socket(session->parameter);
    ...
    /* start by blasting out every block */
    xfer->block = 0;
    while (xfer->block <= param->block_count) {
        ...
        /* see if transmit requests are available */
        status = read(session->client_fd, ((char*)&retransmission)+
        retransmitlen, sizeof(retransmission)-retransmitlen);
        ...
        /* if we have no retransmission */
        } else if (retransmitlen < sizeof(retransmission_t)) {
            /* build the block */
            xfer->block = min(xfer->block + 1, param->block_count);
            ...
            status = build_datagram(session, xfer->block, block_type,
datagram);
            ...
            /* transmit the block */
        status = sendto(xfer->udp_fd, datagram, 6 + param->block_size,
        0, xfer->udp_address, xfer->udp_length);
```

<그림 13> 대용량 데이터를 서버 측으로 전송하는 클라이언트의 소스코드

상기 과정을 거쳐 클라이언트로부터 서버 측으로 데이터가 전달된다. 서버 측에서는 이를 수신하여 디스크에 저장함에 있어 파일명을 알고 있어야 한다. 이는 실제적 데이터 전달을 위한 제어 정보에 해당하는 것으로서 <그림 12>에서 생성한 TCP 소켓을 파일 포인터 형태로 변환하였고 이를 통해 클라이언트와 서버 간의 원활한 제어 정보 전달이 가능하다. 파일명을 비롯 데이터 블록 사이즈, 블록 카운트, 등의 전송 관련 각종 정보는 별도의 구조체에 저장되며 이를 통해 클라이언트와 서버 간의 체계적인 파일 전송 구현이 가능하다. 서버 상에서 파일 저장을 위해 wb를 옵션으로 하는 fopen 함수 호출을 통해 파일을 생성하였고 클라이언트로부터 수신되는 데이터를 이곳에 저장하기 위해 UDP 소켓을 개설하였다. 수신 과정에서 비연결형 UDP로 인한 패킷 손실을 보상하기 위해 재전송 테이블을 생성하였을 뿐만 아니라 링 버퍼 및 디스크 I/O를 통한 원활한 파일 수신 처리를 위해 링 버퍼 및 디스크 I/O 쓰레드를 지정하였다. 클라이언트와 마찬가지로 데이터 수신을 위하여 별도로 개설된 UDP 소켓 상에서 recvfrom 함수 호출을 통해 데이터 수신을 구현하였고 블록 단위로 데이터가 수신될 때마다 블록 넘버를 증가시켜 수신 완료를 결정하도록 하였다. 상기 메커니즘을 기반으로 데이터 수신을 위한 서버의 의사 코드를 간략히 나타내보면 아래와 같다.

```
    /* convert our server connection into a stream */
    session->server = fdopen(parameter->server_fd, "w+");
    ...
    /* Effective data receiving and management on server system by
    using data structure*/
    if (fread(filename, MAX_FILENAME_LENGTH, 1, session->server) < 1)
                return warn("Could not read file name");
    xfer->local_filename=strdup(filename);
    if (fread(&xfer->file_size,   8, 1, session->server) < 1)return warn("Could
    not read file size");
          xfer->file_size    = ntohll(xfer->file_size);
    ...
    /* try to open the local file for writing */
    xfer->file = fopen(xfer->local_filename, "wb");
...
    /* open a new datagram socket */
    session->transfer.udp_fd = create_udp_socket(session->parameter);

    /* allocate the retransmission table */
    rexmit->table    =    (u_int32_t    *)    calloc(DEFAULT_TABLE_SIZE,
    sizeof(u_int32_t));
    /* start up the disk I/O thread */
    status = pthread_create(&disk_thread_id, NULL, disk_thread, session);
  while (1) {
      /* try to receive a datagram */
      status = recvfrom(xfer->udp_fd, local_datagram, 6 + session->
      parameter->block_size, 0, NULL, 0);
      ...
      /* keep statistics on received blocks */
      xfer->stats.total_blocks++;
      if (this_type != TS_BLOCK_RETRANSMISSION) {
      ...
```

<그림 14> 클라이언트로부터 대용량 데이터를 수신 처리하는 서버

6. 데이터 전송 테스트 및 성능 분석

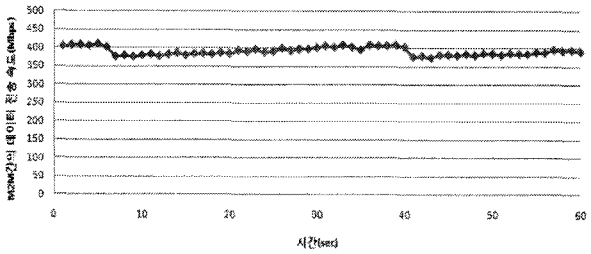
VLBI 각 사이트에서 대용량 데이터를 전송할 수 있는 프로그램 설계 및 개발에 대해 지금까지 살펴보았다. 본 절에서는 초고속 네트워크 상에서 해당 프로

그램의 실제 활용을 통해 개발된 프로그램의 성능, 유효성을 검증하고자 한다. 나아가 데이터 전송 결과에 대한 심층 분석을 통해 보다 나은 성능을 이끌어낼 수 있는 방안에 대해 논의하기로 한다.

6.1 데이터 전송 실험

본 논문에서 개발한 프로그램은 각 VLBI 관측소에서 획득한 대용량 관측데이터를 보다 신속하고 효율적으로 상관센터로 전송하기 위한 목적으로 개발되었다. 각 VLBI 관측소로 수신되는 데이터는 A/D 변환을 거쳐 Mark5B라 불리는 시스템 상에 1Gbps 속도로 기록되며 통상 한번의 천체 관측시 수 TB의 데이터가 양산된다. 얻어진 데이터에 대한 처리 및 분석은 데이터센터에 해당하는 상관센터에서 이루어지며, 안정적이고 신속한 데이터 전송을 위해 KVN의 각 사이트는 국가과학기술연구망 KREONET으로 네트워크 구성되었다. 데이터 전송과 수신이라는 네트워크 관점에서 VLBI 관측소와 상관센터는 각각 클라이언트와 서버에 해당한다. 따라서 개발된 프로그램은 데이터 처리를 수행하는 각 사이트의 컴퓨터 시스템 상에서 실행되어 클라이언트와 서버로서 그 역할을 수행하게 된다.

네트워크 상에서 데이터 전송을 테스트함에 있어 먼저 파악되어야 할 것으로 현 클라이언트와 서버 간의 성능을 들 수 있다. 여기에는 비단 네트워크만이 아닌 데이터 전송을 수행하는 클라이언트와 서버 시스템의 성능도 포함하며, 이에 대한 명확한 측정을 위해 iperf 툴을 활용하였고, 그 결과를 도시하면 아래와 같다.



<그림 15> 메모리 vs 메모리 간의 네트워크 전송 성능

위 그림은 클라이언트와 서버 시스템을 포함한 네트워크 상에서 UDP 기반의 전송 성능을 도시한 것으로서 메모리 vs 메모리를 기준으로 한다. TCP가 아닌 UDP 기반의 데이터 전송이기에 패킷 손실율을 동반하지만 그를 0.96%로 최소화하였고 388Mbps의 전송 성능을 얻을 수 있었다. 본 논문에서는 이러한 성능의 네트워크 상에서 디스크 vs 디스크 간의 효율적인 데이터 전달을 구현하고자 하며 그에 대한 평가를 위해 dd 명령을 통해 1GB 크기의 샘플 파일을 별도로 생성하였다. 또한 네트워크 전송 과정에서 발생할지 모르는 패킷 유실로 인한 데이터 변조 유무를 판변할 수 있도록 클라이언트는 MD5 체크섬을 서버 측으로 함께 전송하도록 하였다[9]. 파일 수신 후 서버는 해당 MD5 체크섬을 전송 완료된 파일에 적용하여 파일의 이상 유무를 검증할 수 있도록 하였는데 아래에서 그에 대한 결과를 나타내었다.

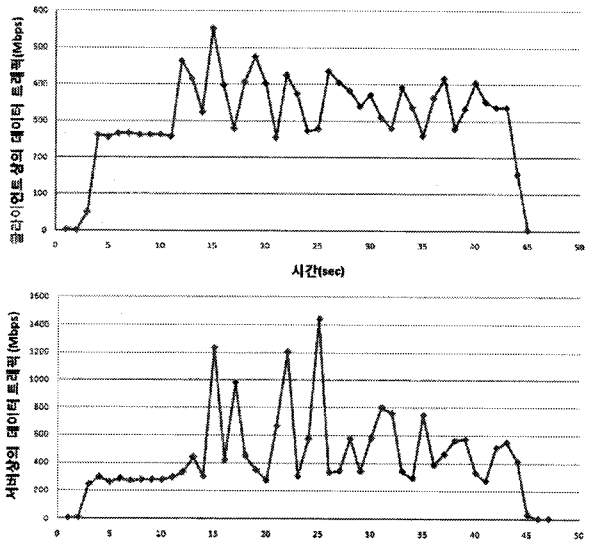
```
Client:/home/oper/work/file_client# ./client 203.250.155.110 1GB
Client is connected to remote server successfully!!
Block size: 1024
Buffer size: 20000000
Port: 46224
File name transferred to server: '1GB'
                    last_interval                                  transfer_total
              buffers        transfer_remaining   OS UDP
time              blk     data          rate rexmit      blk     data
  rate rexmit queue  ring     blk   rt_len      err
00:00:00.464  300    0.00M   161.4Mbps   0.0%      300     0.0G
161.4Mbps   0.0%      0     2    31700        0        0 --
00:00:00.831  350    0.00M   238.9Mbps   0.0%      650     0.0G
195.5Mbps   0.0%      0     1    31350        0        0 --
00:00:01.184  350    0.00M   247.9Mbps   0.0%     1000     0.0G
211.1Mbps   0.0%      0     1    31000        0        0 --
...
Transfer complete. Flushing to disk and signaling server to stop...
!!!!
PC performance figure : 0 packets dropped (if high this indicates
receiving PC overload)
Transfer duration      : 39.88 seconds
Total packet data      : 9239.25 Mbit
Goodput data           : 7892.75 Mbit
File data              : 8000.00 Mbit
Throughput             : 257.66 Mbps
Goodput w/ restarts    : 123.90 Mbps
Final file rate        : 226.4 Mbps
Transfer mode          : lossless
Client:/home/oper/work/file_client#
```

```
Server:/home/oper/work/file_server% ./server
Server started.
Waiting for client..
Client is connected to server successfully!!
 erate      ipd  target    block    %done srvNr
     0 1007.50us    403us      311     0.97    0
```

```
     0 839.58us   403us    663   2.07   0
     0 699.65us   403us   1015   3.17   0
     0 583.04us   403us   1414   4.42   0
     0 485.87us   403us   1815   5.67   0
...
Transmission complete.
Server 0 transferred 1048576000 bytes in 41.82 seconds (200.59 Mbps)
Server:/home/oper/work/file_server% md5sum --check 1GB.md5
1GB: 성공
Server:/home/oper/work/file_server%
```

<그림 16> 클라이언트 및 서버 상의 데이터 송수신 내역

클라이언트 시스템의 디스크에 저장된 1GB 크기의 파일을 네트워크를 통해 서버 상의 디스크로 전송함에 있어 위로부터 대략 230Mbps의 데이터 전송 속도를 달성할 수 있었고 MD5 체크섬 수행을 통해 해당 파일의 무결성도 확인할 수 있었다. 데이터 전송과 관련하여 각 시스템의 네트워크 인터페이스 카드에서 입출력되는 트래픽을 그래픽 형태로 <그림 17>에 도시하였다.



<그림 17> 각 시스템 NIC에 발생하는 트래픽 모니터링

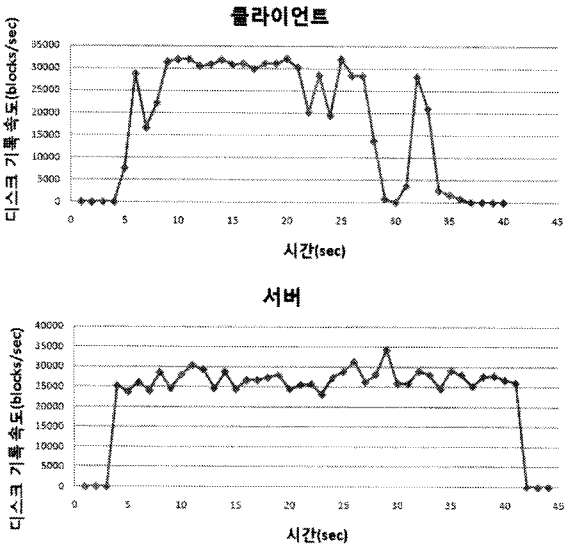
이는 4.1 섹션에서 소개한 VTP를 이용한 데이터 전송 방식보다 월등히 개선된 성능으로서 본 논문에서 개발한 Tsunami-UDP 기반의 애플리케이션을 통해 2배 이상의 성능 향상 효과를 얻을 수 있었다. 뿐만 아니라 MD5 체크섬 적용을 통해 대용량 데이터 전송에 있어서 QoS 제고는 물론, 데이터 전송에 있어 패킷 유실이 발생하지 않았음을 검증할 수 있었다.

6.2 성능 분석

본 논문에서 대용량 데이터 전송을 위해 사용하는 네트워크는 국가과학기술연구망인 KREONET으로서 최상의 성능과 안정도를 보장하는 네트워크이다. 양질의 네트워크 환경이기에 일반 공용망과 같은 여러 사용자 간의 트래픽 유발 및 그로 인한 네트워크 혼잡을 최소화할 수 있다. 이에 따라 TCP가 아닌 UDP를 전송 프로토콜로 지정하였고 오픈소스인 Tsunami-UDP 프로그램을 KVN의 상황에 맞게 업그레이드 하였다. 이러한 환경에서 실제 데이터 전송 시험을 통해 KVN 관측소와 상관센터 간에 230Mbps에 달하는 데이터 전송 성능을 확인할 수 있었다. 일반 기업망 및 상용망에 비하면 월등한 수치이기는 하지만 이는 하루에 수 TB의 관측 데이터를 송수신해야 하는 KVN의 관점에서 바라보았을 때 미흡한 결과에 해당한다. 하지만 네트워크를 기준으로 한 데이터 전송 성능은 이보다 높게 나왔으며, 6.1 섹션에서 그에 대해 확인할 수 있었다. 이는 네트워크가 월등한 성능을 제공하고 데이터 전송에 특화되어 있다 하더라도, 그에 연결되어 있는 시스템의 사양이 낮거나 최적화되어 있지 못하다면 병목현상이 발생할 수 있다는 것을 나타내는 것으로서, 네트워크와 더불어 시스템 업그레이드의 중요성을 암시하고 있다.

네트워크 기술의 급격한 발전에 따라 현재 광케이블을 통해 전송 가능한 데이터 속도는 하드디스크 드라이브의 그것을 월등히 상회하고 있다. Gbps 이상의 데이터 전송을 네트워크 환경이 지원하더라도 그를 송수신하는 컴퓨터 시스템이 그에 부합되지 못하면 원하는 속도의 데이터 전송은 불가하다. 더욱이 그것이 각 시스템의 메모리 사이의 전송이 아니라 디스크 간의 송수신이라면 성능은 더욱 저하될 수 밖에 없다. 이와 관련하여 데이터 전송 과정에서 각 시스템에 발생한 디스크 I/O를 vmstat 툴을 활용하여 모니터링한 결과를 도시해보면 아래와 같다.

해당 그림 상의 그래프는 매 초당 디스크에 입출력된 데이터 크기를 1KB 블록 단위로 나타낸 것으로서



<그림 18> 각 시스템의 디스크 입출력 성능

각각 30000, 5000 blocks/s에서 수렴하고 있다. 이는 bps 단위로 환산했을 때 클라이언트와 서버 각각 대략 240, 210 Mbps의 기록 속도를 의미하는 것으로서 6.1 섹션의 데이터 전송 속도와 대략 일치함을 알 수 있다. 이는 본 논문에서 개선한 Tsunami-UDP 기반의 프로그램이 보다 높은 속도의 데이터 전송을 구현할 수 있으나 엔드단의 시스템 사양으로 인해 입출력되는 데이터가 블록되는 것을 의미하는 것으로 <그림 17, 18>의 비교를 통해서도 분명히 알 수 있다. 클라이언트, 서버 시스템의 디스크 입출력 특성이 나타난 <그림 18>의 경우 다소 변동은 있으나 특정의 수치로 수렴하는 것을 알 수 있다. 하지만 각 시스템의 NIC로 입출력되는 데이터 트래픽의 경우 지그재그 형태로 일정치도 않고 상당한 변화를 보이는 것을 알 수 있다. 이는 네트워크를 통해 빠른 속도로 데이터가 전달되나, 그를 송수신하는 말단 시스템의 성능이 네트워크 성능을 따라가지 못하여 발생하는 현상으로서 이미 <그림 15>에 도시한 메모리 vs 메모리 단의 데이터 전송 성능을 통해서도 분명히 확인할 수 있다.

7. 결 론

e-VLBI는 네트워크를 통해 대용량 VLBI 데이터를 전송하기 위한 e-Science의 한 연구 분야로서 각 관측소에서 Gbps급의 속도로 유입되는 데이터를 상관센터로 실시간 전송하는 것을 그 궁극적 목표로 한다. 국내에서는 KVN을 중심으로 e-VLBI 연구에 대한 연구가 현재 진행되고 있으나 본격적인 연구가 시작된 지 아직 얼마 되지 않았기에 일차적으로 대용량 디스크에 기록된 데이터를 상관센터로 초고속으로 전송하는 것을 목표로 하고 있다. 이에 따라 본 논문에서는 VLBI 관측소에서 획득한 대용량 관측 데이터를 네트워크를 경유하여 효과적으로 전송할 수 있는 프로그램에 대한 개발, 전송 시연 및 성능 분석에 대해 논의하였다. 데이터 전달에 있어 TCP의 경우에는 AIMD(Aditive Increase, Multiplicative Decrease) 방식의 혼잡 제어 알고리즘에 기반하기에 전송 효율성이 저하될 수 밖에 없다. 이에 따라 본 논문에서는 데이터 전송을 위한 기반 프로토콜로서 비연결형 데이터그램 전달 방식의 UDP를 활용하였고, Tsunami-UDP를 기본 알고리즘으로 하여 대용량 데이터 전달 프로그램을 구현하였다. VLBI 관측소와 상관센터를 각각 클라이언트와 서버로 지정하여 개발된 프로그램을 실행하였고 각 사이트의 컴퓨터 상에서 실행하였고 실제 데이터 전송 시연을 통해 VTP를 활용한 기존의 VLBI 데이터 전송 방식에 비하여 월등히 향상된 전송 효율을 달성할 수 있었다. 하지만 메모리 vs 메모리 방식으로 측정한 네트워크의의 성능과 비교하였을 때 실제 데이터 전송 성능은 그에 미치지 못하였다. 이에 대한 심도있는 성능 분석을 통하여 본 논문에서는 디스크를 비롯한 컴퓨터 디바이스가 데이터 전송을 지연시키는 요인으로 작용함을 알 수 있었고, 데이터 전송 과정에서 NIC와 디스크의 입출력을 모니터링하여 그에 대한 근거를 확보할 수 있었다.

대용량 데이터의 실시간 또는 준-실시간 전송은 VLBI 연구에 있어 절대적으로 중요한 부분이며, 향후 그에 대한 필요성은 더욱 커질 것으로 예상된다. 대용량 데이터 전달에 있어 전제되어야 할 것으로 높은 QoS를 갖는 첨단망, 높은 전송 효율의 프로토콜과 함께 최적화된 컴퓨터 시스템을 들 수 있다. 이에 따라 VLBI 관측소 및 상관센터 내의 데이터 송수신 시스템에 대한 업그레이드 및 적절한 튜닝이 필요하다고 생각하며, 이러한 과정을 거쳐 보다 완성도 높은 e-VLBI 시스템 아키텍처를 구현할 수 있을 것으로

생각한다.

Acknowledgement

이 연구는 한국과학기술정보연구원에서 운영하는 국가과학기술연구망(KREONET)에 기반하여 수행되었습니다.

참 고 문 헌

[1] Fran Berman1 et al, "Grid Computing: Making the Global Infrastructure a Reality", John Wiley & Sons, Ltd, pp.780-787, 2003.

[2] Alan R. Whitney, Dan L. Smithe and John A. Ball, "Mark5B DIM command set(Revision 1.12)", MIT Haystack Observatory, pp.1-10, 2006.

[3] Thomson A. R., Moran J. M., & Swenson G. W. Jr.,"Interferometry and Synthesis in Radio Astronomy" NRAO Socorro Observatory, pp.312-335, vol. 2, 2001,

[4] 손봉원, 송민규, "e-VLBI e-초장거리간섭계", 2007년도과학기술연구망워크숍, 2007.

[5] 송민규 외, "e-VLBI 구현을 위한 네트워크 모델 설계", 천문학 논총, 제 20권, 제1호, 2005..

[6] KREONET 홈페이지 (http://www.kreonet.re.kr/01_info/01_info.php)

[7] Martin, J., Nilsson, A., Injong Rhee, "Delay-based congestion avoidance for TCP", Networking, IEEE/ACM Transactions on Networking, vol. 11, issue 3, pp. 356-369, 2003.

[8] Mark R. Meiss, "Tsunami: A High-Speed Rate-Controlled Protocol for File Transfer", Indiani University, pp.1-10, 2003.

[9] Sherwood, R., Braud, R., Bhattacharjee, B., "Slurpie: a cooperative bulk data transfer protocol", INFCOM, vol.2, pp.941 - 951, 2004.

송 민 규 (Min-Gyu Song)

- 정회원
- 강원대학교 전기공학과 공학사
- 강원대학교 전자공학과 공학석사
- 한국천문연구원 기술개발연구본부 연구원

- 관심분야 : 초고속네트워크, 분산 컴퓨팅, 리눅스 프로그래밍

김 현 구 (Hyun-Goo Kim)

- 서울대학교 천문학과 이학사
- 서울대학교 천문학과 이학석사
- 서울대학교 천문학과 이학박사
- 한국천문연구원 전파천문연구본부 대덕전파천문대장

- 관심분야 : 전파천문학, e-Science

손 봉 원 (Bong-Won Sohn)

- 연세대학교 천문대기학과 이학사
- 연세대학교 천문대기학과 천문대기학석사
- 독일 Bonn대학교 천문학과 천문학박사

- 한국천문연구원 전파천문연구본부 선임연구원, 한국과학기술연합대학원 천문우주학과 부교수(겸임)
- 관심분야 : VLBI 정밀계측, e-VLBI

위 석 오 (Seog-Oh Wi)

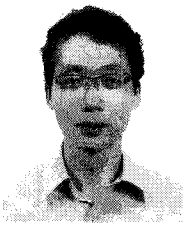
- 전남대학교 전기공학과 공학사
- 전남대학교 전기공학과 공학석사
- 전남대학교 전기공학과 공학박사
- 한국천문연구원 기술개발연구본부 한국천문연구원

- 관심분야 : 전파망원경제어, 제어계측, 전력전자

강 용 우(Yong-Woo Kang)

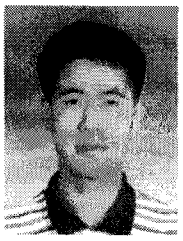
- 부산대학교 기계설계학과 공학사
- 부산대학교 지구과학과 이학석사
- 부산대학교 지구과학과 천문학전공 이학박사

- 한국천문연구원 기술개발연구본부 선임기술원
- 관심분야 : 전파백엔드시스템, 대용량 자료처리, 관측천문학



염 재 환 (Jae-Hwan Yeom)

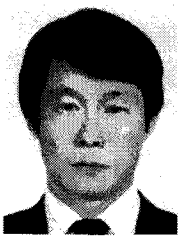
- 한국천문연구원 전파천문연구본부 상관기연구그룹 선임기술원
- 관심분야 : 우주전파신호처리, 고속상관처리, 초고속데이터전송시스템



변 도 영 (Do-Young Byun)

- 서울대학교 천문학과 이학학사
- 서울대학교 천문학과 이학석사
- 서울대학교 지구환경과학부 이학박사

- 한국천문연구원 전파천문연구본부 선임연구원
- 관심분야 : 전파천문학



한 석 태 (Seog-Tae Han)

- 한양대학교 무선통신공학과 공학사
- 광운대학교 전자통신공학과 공학석사
- 충남대학교 전자공학화 공학박사

- 한국천문연구원 기술개발연구본부 책임연구원, 기술개발연구본부장
- 관심분야 : mm/sub-mm파 우주전파수신기 개발, Quasi-Optics 설계 및 제작