

# 우주 패킷 통신에서 우주 통신 네트워크로의 문을 여는 LTP 프로토콜 구현

구 철 회\*

## An Implementation of LTP Protocol Opening a Gate to Space Communication Network from Space Packet Communication

Cheol Hea Koo\*

요 약

1957년 소련이 스푸트니크 1호를 최초로 우주로 발사한 이후 인류는 무수한 지구궤도 위성을 발사하였으며 심지어 달과 화성에 궤도선과 착륙선을 보내고 태양계 영역 넘어까지 탐사선을 보냈다. 우주 개발 초기부터 탐사선과 지상국간에는 직접 통신 방식이 사용되어 왔다. 심우주 통신의 경우 통신 거리가 지구-화성간 거리와 같이 늘어나고 각 국 및 각 기관간 우주탐사 임무가 서로 협력하는 경우가 늘어남에 따라 통신 중계의 기술적 요구가 증대되고 있어 중단허용네트워크 기술을 근간으로 한 태양계 통신 네트워크의 시대가 문을 열고 있다. 본 논문에서는 중단허용네트워크 프로토콜의 점대점 통신 프로토콜인 LTP 표준을 구현한 연구내용 및 이의 의미를 발표하였으며 점점 우주통신 네트워크 기술 성장에 따라 향후 지연 허용 네트워크 및 우주 패킷 프로토콜 간 중계 구현 활용에 기여하고자 한다.

**Key Words** : Licklider Transmission Protocol, Convergence Layer, Space Packet Protocol, Delay Tolerant Networking, Communication Relay, Deep Space, Consultative Committee for Space Data Systems, space exploration, bundle protocol

ABSTRACT

Since the Soviet Union launched the first Earth's satellite, Sputnik I, in 1957, numerous satellites have launched at the Earth orbit. Human being has reached outer space beyond Earth, Moon, Mars, and even the outer solar system by sending orbiters, landers, and rovers. Direct communication between spacecraft and ground stations have dominated as a primary way to talk to space until communication distance was not severely challenged and global cooperation of space exploration between nations and space agencies was not frequently requested. Now, the solar system internetworking era has opened for coping with the challenges in deep space exploration by adapting the delay-/disruption-tolerant networking (DTN) technology. This paper presents the result of the implementation of Licklider Transmission Protocol (LTP) as a key protocol preparation of the DTN protocol software suite. Using the LTP in space packet communication provides a suitable solution to remedy disruptions from the harsh space communication environment. As space communication network evolves, this research may contribute as one of the major prerequisites during the construction of the delay-tolerant network in deep space as space assets needs to interconnect to the solar system internetwork in the future.

\* 본 연구는 한국항공우주연구원 기본사업(FR21M00) 연구의 일환으로 수행되었습니다.

\* First and Corresponding Author : Korea Aerospace Research Institute, chkoo@kari.re.kr, 정희원

논문번호 : 202108-211-A-RN, Received August 20, 2021; Revised October 13, 2021; Accepted October 20, 2021

## I. 서 론

수천년전 고대 인류가 수신호 또는 고대 원시 언어로 대화를 시작한 이후로 인류는 단체 생활을 시작하게 되었다<sup>[1]</sup>. 통신 수단의 발달은 인류 공동 활동의 규모가 소규모 부족에서 도시, 국가로 확장되는데 매우 큰 기여를 하였다. 이는 우주 통신에서도 크게 다르지 않다. 1957년 소련이 스푸트니크 1호를 최초로 우주로 발사한 이후 인류는 무수한 지구궤도 위성을 발사하였으며 심지어 달과 화성에 궤도선과 착륙선을 보내고 태양계 영역 넘어까지 탐사선을 보냈다. 우주 개발 초기부터 탐사선과 지상국간에는 직접 통신(direct communication) 방식이 사용되었는데 각 기관 또는 국가간 상이하던 통신 프로토콜은 국제 표준화 논의를 거쳐 CCSDS(Consultative Committee for Space Data Systems, 우주데이터시스템 자문위원회) 및 ESA PSS(European Space Agency Procedures, Specifications and Standards, 유럽우주국 프로시저, 규격 및 표준) 패킷 통신 프로토콜(SPP, space packet protocol, 우주패킷프로토콜)<sup>[2]</sup>로 정립되었으며 현재 수행되고 있는 우주통신 방식의 대다수가 이를 사용하고 있다. SPP는 점대점(point-to-point) 통신 프로토콜로서 통신 중계 기능은 포함하지 않고 있다. 심우주 통신의 경우 통신 거리가 지구-화성간 거리와 같이 늘어나고 각 국 또는 각 기관간 우주탐사 임무가 서로 협력하는 경우가 늘어남에 따라 통신 중계의 기술적 요구가 증대되고 있다. 그 대표적인 예가 화성 궤도선인 MRO(Mars Reconnaissance Orbiter, 미국, 2006년 발사)와 Mars Express(ESA, 2003년 발사)가 화성 표면 로버와 지구간 임무 데이터 전송과정에 중계위성으로 관여하고 있는 것이다. 비록 MRO와 Mars Express가 통신 중계에 관여하고 있지만 그 방식은 자동화된 데이터 통신 네트워킹하고는 다른 방식이다. 따라서 상기 통신 중계 기능은 직접 전달(bent-pipe) 방식에 기반한 매우 제한적이고 특수한 상황에 대비하도록 구성되어졌다. BER(bit error rate)가 높을수록 직접 전달 방식에 비해서 DTN 라우팅 방식의 데이터 전달율이 높기 때문에 효율적인 심우주 탐사 통신 시스템 구축에 DTN은 장점을 부여한다<sup>[3]</sup>.

심우주 통신 환경에 적합한 우주 통신 네트워킹 기술을 심우주 탐사 임무에 적용하기 위한 연구가 미국을 중심으로 연구되어 왔으며 2005년 소행성 탐사를 위해 발사한 EPOXI(Extrasolar Planet Observation and Deep Impact Extended Investigation) 임무의 일환으로 수행된 DINET(Deep Impact DTN

Experiment) 실험을 통해 그 실체가 세상에 처음으로 소개되었다<sup>[4]</sup>. 이 실험을 통해서 나타난 태양계 인터넷(Solar System Internet, SSI) 또는 항성간 네트워크(Interplanetary Network, IPN)이라고 부르는 우주 통신 환경에서 사용될 목적으로 고안된 네트워킹 기술은 중단허용네트워킹(Delay-/Disruption-Tolerant Networking, DTN) 프로토콜로 불리게 되었다<sup>[5]</sup>. DTN 프로토콜은 지상 인터넷(terrestrial internet) 기술과 유사하며 기본적으로 OSI 7 계층에 따라서 각 구성계층을 분리할 수 있다. 이중 통신 하드웨어를 제외한 프로토콜적 요소를 갖고 있는 부분은 데이터 링크 계층과 네트워크 계층으로 압축할 수 있는데 네트워크 계층은 번들 프로토콜<sup>[6]</sup>을 사용하며 지상 인터넷의 라우팅 프로토콜과 그 궤를 같이한다. 통신 지연이 길고 비트 오류 확률이 높은 우주 통신 환경의 특성상 데이터 링크 계층은 융합 계층(convergence layer)으로 통상 호칭되며 지상 인터넷의 TCP/IP 프로토콜과 대비된다. 지상 인터넷 노드간 신뢰성 연결이 TCP/IP 프로토콜로 수립되는 것과 같이, 하나의 우주통신 자원이 우주 통신 네트워크와 신뢰성있게 연결되기 위해서는 융합 계층 프로토콜이 지원되어야 한다. 본 논문에서는 DTN 프로토콜에서 우주통신 지원간 융합 계층 프로토콜 표준으로 자리잡은 LTP(Licklider Transmission Protocol)<sup>[7]</sup> 프로토콜을 분석하고 개발 내용 및 타 구현 소프트웨어와 호환성 시험을 수행한 결과를 발표할 예정이다. 결론 부분에서는 본 개발의 의의와 향후 활용 방안에 대해서 언급한다.

## II. 융합 계층

그림 1에는 우주통신에서 구성될 수 있는 DTN 프로토콜 계층도를 도시하였으며<sup>[8]</sup> 그림에서 보여지는 것과 같이 융합 계층은 링크 계층과 결합하여 DTN 프로토콜을 지원한다. 근본적으로 융합 계층은 어떤 링크 계층 프로토콜과도 협력하여 작업을 처리할 수 있다. 이는 반대로 그만큼 우주 환경에서 링크 계층의 통일화가 어렵다는 반증도 된다. 지상 환경에서는 TCP/IP, UDP/IP 방식의 링크 계층이 사용될 수 있으나 우주 환경에서는 긴 통신 지연(long propagation delay 또는 long round-trip time), 높은 통신 오류(high bit-error rate), 비대칭적 통신 대역폭(asymmetrical communication bandwidth) 등의 특징 때문에 지상 인터넷 기술을 그대로 적용할 수 없다.

융합 계층의 기능은 DTN(특히 번들 계층) 프로토콜과 링크 계층 사이의 신뢰성 데이터 교환이며 긴 통

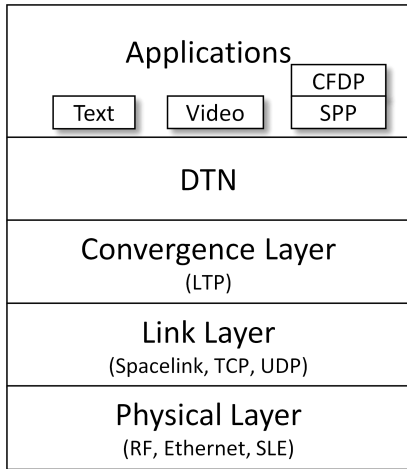


그림 1. 우주통신의 DTN 프로토콜 계층도  
Fig. 1. Layer of DTN protocol in space communication.

신 지연 및 높은 BER, 비대칭적 통신 속도 조건에서도 신뢰성있는 데이터 송수신을 지원할 수 있도록 고안되었다. 특히 특정 전송 매체 및 통신 방식에 얽매이지 않는 추상화된 통신 정보 제어 규칙에 기반을 두고 있기 때문에 다양한 우주 임무 및 탐사 환경에서 적용할 수 있다는 특징을 가지고 있다고 할 수 있다. 따라서 실 사용시에는 LTP over Spacelink, LTP over TCP/IP, LTP over UDP/IP 방식 등으로 운용되며 지상 시험 환경에서는 대표적으로 UDP/IP 상에서 시험 되도록 구성되고 있다<sup>9)</sup>. 우주상에서는 고전적인 RF 링크를 비롯하여 Proximity-1 링크와 레이저 광통신 채널과도 연결이 가능하다. 실제로 2013년 발사된 달 탐사 궤도선 LADEE(Lunar Atmosphere and Dust

Environment Explorer, 미국 NASA ARC)의 LLCD (Lunar Laser Communication Demonstration) 광통신 탑재체를 이용하여 시험되었다<sup>10)</sup>.

### III. LTP 표준 및 개발

LTP 표준은 RFC-5326 및 CCSDS 734.1에 기술되어 있다. LTP의 핵심은 Ack 신호에 의한 재전송(retransmission)에 기반하고 있다는 점에서 TCP/IP 프로토콜과 유사하지만 원래 순서대로의 전달(in-order delivery)을 지원하지 않는 점에서는 다르다. 즉 성공적으로 받지 못한 부분에 대해서 재전송을 요청하여 완전한 데이터 수신을 보증하지만 원래 순서대로의 전달은 보장하지 않는다.

LTP 표준 규격은 큰 흐름의 프로토콜 정의는 하고 있지만 상세한 내부 동작은 정의하지 않고 있기 때문에 구현자 재량(implementation discretion)에 따라서 운용 방법이 차이가 날 수 있다. 디버그와 개발 용이성 등으로 인해 LTP 표준 개발 플랫폼으로 x86 프로세서 기반 리눅스 운영체제를 선택하였다. 그림 2에는 LTP 엔진의 개념도를 나타내었다. RF 방식으로 전달된 데이터는 소프트웨어 정의 라디오(Software Defined Radio, SDR) 장치를 통해서 접속된다. LTP 통신 모듈은 LSI(link service input)와 LSO(link service output) 포트를 통해서 외부와 접속하게 되며 지상 실험에서는 UDP 통신을 사용한다<sup>9)</sup>.

아래에는 LTP 구현의 핵심적인 부분 및 주요 핵심 결정 사항을 기술하였다.

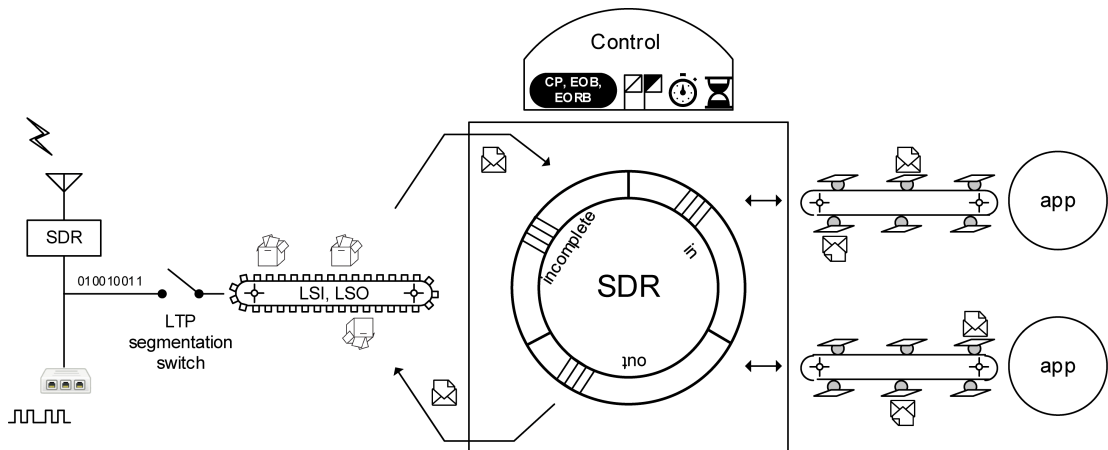


그림 2. LTP 엔진 모델 개념도  
Fig. 2. Conceptual diagram of LTP engine model.

### 3.1 SDR

SDR(spacecraft data recorder 또는 simplified data recorder)는 LTP 세그멘테이션 데이터를 임시적으로 보관하는 요소이다. 자동차가 다니는 도로를 닦는 것이 SDR이라고 하면 교차로에서 신호기에 해당하는 것이 LTP 프로토콜이라고 할 수 있다.

SDR은 임의의 위치에서 세그멘테이션 데이터의 삽입-삭제-이동이 자유로운 이중 연결 리스트(doubly linked list) 데이터 구조체를 핵심 요소로 하고 이를 LTP 처리 엔진이 편리하게 사용할 수 있고 데이터 완결성을 위해 상호 배제(mutual exclusion) 접근성을 제공하는 유틸리티를 제공한다.

CCSDS에서는 그림 3과 같은 단순한 LTP 아키텍

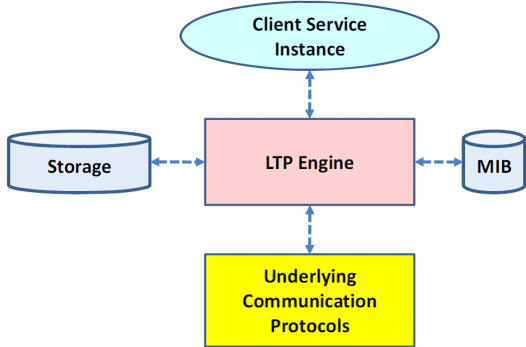


그림 3. LTP 아키텍처 구성 요소  
Fig. 3. Architecture element of LTP.

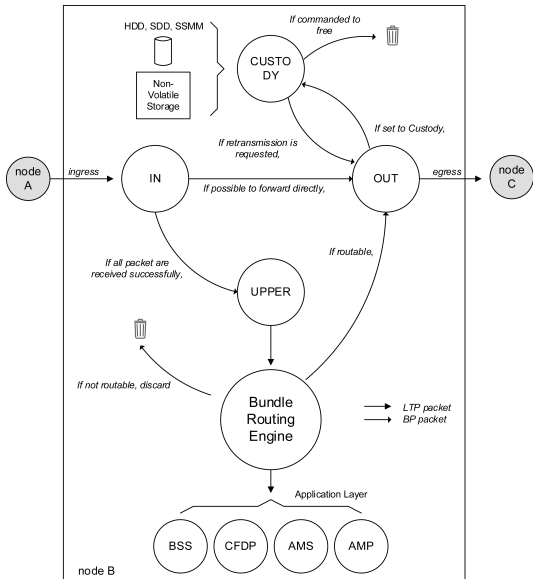


그림 4. DTN 노드에서의 SDR 운용 개념도  
Fig. 4. Operation concept of SDR in a DTN node.

처를 제안하고 있다<sup>9)</sup>.

DTN에서는 LTP 세그멘테이션 데이터 자체가 번들 데이터이기 때문에 SDR은 LTP 동작뿐만 아니라 번들 데이터 처리에도 같이 사용된다. 그림 4에는 일반적인 DTN 노드에서 SDR의 동작 개념을 모사하였다.

오류없이 수신된 세그멘테이션 데이터는 그림 2와 같이 우선 미완성(incomplete) SDR 영역에 보관되어 추후 관련 상황 및 조건에 따라서 수신(in) SDR 영역으로 옮겨지도록 하였다. 수신 SDR 영역으로 옮겨진 세그멘테이션 데이터는 즉시 해당 유저 응용 프로그램으로 보내진다.

### 3.2 Thread model

LTP 엔진 소프트웨어의 복잡도를 최대한 낮추기 위해서 쓰레드(thread)의 구성을 최소로 설계하였다. 쓰레드는 외부 접속(external interface)을 위한 쓰레드, 내부 유저와의 접속(internal user interface)을 위한 쓰레드 등 2개의 쓰레드 풀만 존재하며 LTP 제어 로직은 세그멘테이션 데이터 수신 즉시 수행되어 SDR 저장시 반영된다. 출력 쓰레드는 송신(out) SDR 영역에 데이터가 있으면 즉시 송출 속도에 맞추어 내 보내게 된다.

그림 5에 LTP의 쓰레드 모델을 나타내었다. 번들 계층과는 BP-UTA(utilization adapter) 접속을 통해서 인터페이스한다. 상호 배제에 의해서 모든 SDR 및 LTP 처리 로직은 쓰레드 안전(thread-safe) 및 재진입 가능(reentrant) API로 동작한다. 처리 로직의 단순화를 위해서 중앙에서 전체적인 흐름을 관장하는 스케줄러(scheduler) 또는 조정자(moderator)는 고려하지 않았다.

### 3.3 SDNV

LTP 표준에서 LTP의 세그멘테이션 데이터 표시에는 최대한 데이터 사이즈를 줄이기 위해서 자체제한 수치표시법(Self-Delimiting Numeric Value, SDNV)을 사용하고 있다<sup>11)</sup>. SDNV 인코딩, 디코딩에 약간의 프로세싱 오버헤드가 있지만 결과적으로 전체 대역폭에서 LTP 고정 헤드가 차지하는 비율을 최소화할 수 있는 효과가 있다.

기존에 한국항공우주연구원에서는 NASA ION의 SDNV 인코딩, 디코딩 함수 개선에 대해 연구한 바가 있으며<sup>11)</sup> 본 LTP 구현에서도 현재까지 가장 최적화된 SDNV 함수를 사용하였다. 다만 차기 LTP 표준에서는 하드웨어 적용성 때문에 가변이 아닌 고정 길이 헤더 필드를 채용하는 것으로 변모하고 있다.

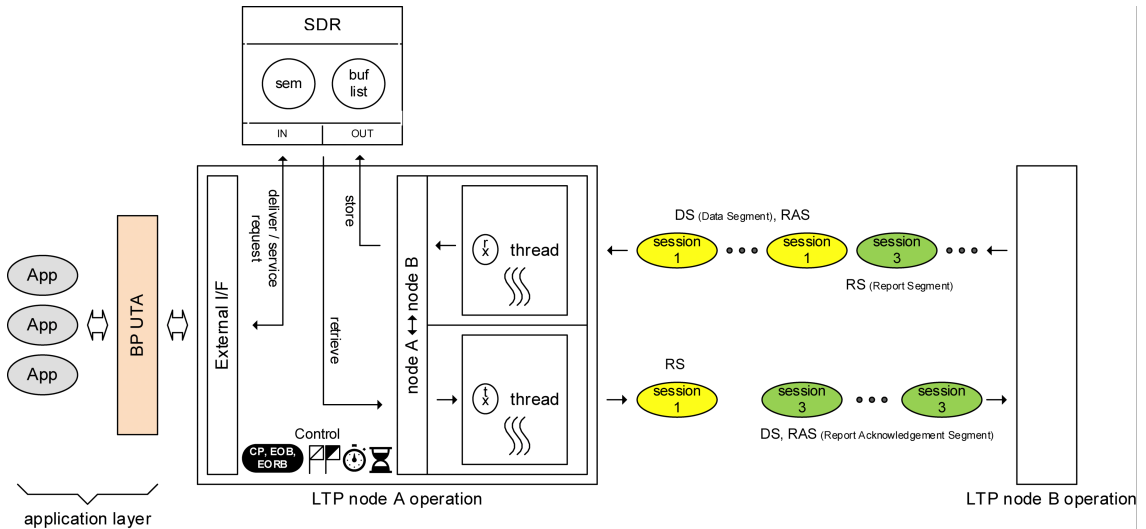


그림 5. LTP 엔진 스레드 모델  
Fig. 5. LTP engine thread model.

### 3.4 구현 및 시험

표 1에는 LTP 구현 환경을 정리하였다. 리눅스 운영체제에서 지원하는 라이브러리 외에는 모두 자체 구현하여 적용하였으며 향후 리눅스 이외의 타 운영체제로 이식 및 적용을 위해서 부가적 라이브러리는 사용하지 않았다. 테스트 함수 역시 자체 개발하여 적용하였다.

표 2에는 LTP 구현시 개발된 주요 소프트웨어 모듈의 함수 리스트를 나타내었다.

표 1. 구현 환경 요약  
Table 1. Summary of Implementation Environment.

Item	Contents
Development Environment	Vmware Workstation Pro 16.x Guest OS : Ubuntu 16.04 x64 LTS
Language	C
Compile	cmake
Thread programming	Posix pthread, Mutex, Condition library
Socket programming	plain linux socket library
Unit test	self-defined test procedure
command script	self-defined test script

표 2. 주요 모듈 함수 리스트  
Table 2. Function list of major module.

Item	Contents
SDR Management	sdrElt_list_alloc()
	sdrElt_list_dealloc()
	sdrElt_list_clear()
	sdrElt_list_move()
	sdrElt_list_first_obj()
	sdrElt_list_last_obj()
	sdrElt_list_next_obj()
	sdrElt_list_prev_obj()
	sdrElt_list_putdata()
	sdrElt_list_putdataOffset()
	sdrElt_list_getdata()
	sdrElt_list_getDataLen()
	sdrElt_list_setDataLen()
	sdrElt_list_getBufPtr()
	sdrElt_list_pool_length()
LTP Administration	sdrElt_list_segment_size()
	sdrObj_create()
	sdrObj_destroy()
	sdrObj_elt_add()
	sdrObj_elt_remove()
	system_initialize()
	system_finalize()
	ltpadmin_initialize()
	ltpadmin_finalize()
	ltpadmin_connection_establish()
ltpadmin_connection_terminate()	

LTP Protocol	ltpadmin_connection_management()
	ltpadmin_session_bind()
	ltpadmin_session_unbind()
	ltpadmin_session_find()
	ltpadmin_session_save()
	ltpadmin_session_load()
	ltpadmin_session_dataPutIncomplete()
	ltpadmin_session_dataMoveToIn()
	ltpadmin_session_save()
	ltpadmin_session_buildRcptClaimContents()
	ltpadmin_session_analysisRcptClaimContents()
	ltpadmin_session_allocSingleSegBlock()
	ltpadmin_session_updateDataLenSingleSegBlock()
	ltpadmin_getNodeObjCtxFromSdr()
	ltpadmin_getNodeObjCtxFromRemoteId()
	ltpadmin_session_outgoingSegmentationData()
	ltpadmin_setHomeId()
	ltpadmin_getHomeId()
	ltpadmin_setRemoteId()
	ltpadmin_getRemoteId()
	ltpadmin_wakeupLtpIntDeliverProcThread()
	ltpadmin_wakeupLtpExtSdrOutReadyProcThread()
	ltp_getSessionNumber()
	ltp_composeDataSegment()
	ltp_processRedData()
	ltp_processGreenData()
	ltp_processRedDataCheckPoint()
	ltp_processRedDataEORP()
	ltp_processRedDataEOB()
	ltp_processGreenDataEOB()
	ltp_processCancelSegment()
	ltp_processCancelAckSegment()
	ltp_processReportSegment()
ltp_processReportAckSegment()	
ltp_analysisSegmentFrame()	
ltp_generateSegmentFrame()	
ltp_isDataSeg()	
ltp_isCP()	
ltp_isEOB()	
ltp_isReportSeg()	
ltp_isReportAckSeg()	
ltp_isCancelSeg()	

ltp_isCancelAckSeg()
ltp_analysisSegmentFrame()
ltp_generateSegmentFrame()

LTP 함수의 시험은 각 API에 대한 단위 시험(unit test) 및 랜덤 입력 이벤트에 대한 부하 시험(load test), 그리고 시나리오 기반의 통합 시험(integration test)에 대해서 수행되었다.

구현된 LTP 엔진 성능의 데모는 그림 6과 같은 테스트 환경을 구성하여 수행하였다. 샘플 동영상 파일<sup>[13]</sup>을 vlc player(Ubuntu에서 sudo apt install vlc로 설치)에서 스트리밍하고 이를 LTP 통신으로 인계받아 최종 수신단 vlc player에서 플레이하여 정상적으로 통신 기능이 동작하는지 확인하였다.

시험 결과는 그림 7에 보였다. 테스트 노트로서 LTP node 2와 node 3을 생성하였다. 그림 7 좌측 (a)에서는 LTP node 2와 3을 각각 초기화하고 스트리밍 전용 vlc player에서 만들어진 동영상 플레이 소스 및 플레이 sink를 준비하였다. 그림 7 우측 (b)에서는 vlc 동영상 플레이어 소스에서 송출한(vlc -dummy -vvv file:///home/xxx/Videos/SV\_1mb.mp4 -sout '#std{access=udp,mux=ts,dst= 0.0.0.0:2114}' 명령 사용) 데이터를 LTP node 2가 수신하여 그림 6의 시험 구성도와 같이 LTP 통신 채널을 이용하여 LTP node 3에 전달하였다. 이 수신된 동영상 스트리밍 소스 데이터를 LTP node 3가 다른 플레이 전용 vlc player에 전달하여 양 노트간에 원활한 영상 플레이(vlc -vvv udp://@localhost: 61000 명령 사용)가 가능한지 확인함으로써 결과적으로 LTP 통신 채널이 정상적으로 동작함을 확인하였다. 그림 7 (b)의 터미널 창에는 LTP 통신 로그가 도시된다.

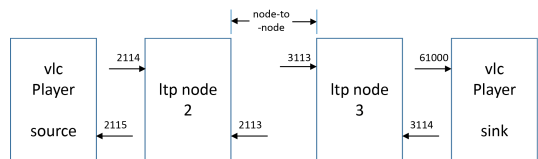
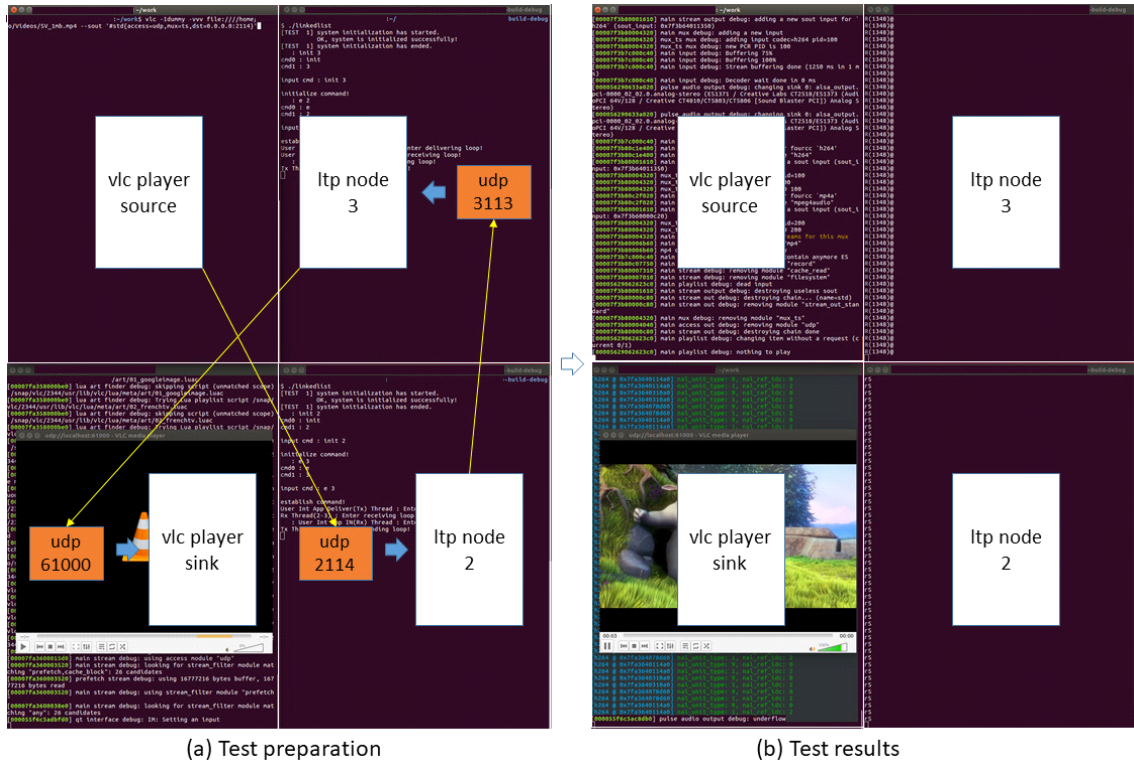


그림 6. LTP 통합 시험 구성도  
Fig. 6. Test configuration of LTP integration test.

### 3.5 호환성 검증

구현된 LTP 기능의 호환성 점검은 이미 검증되어 있는 NASA ION<sup>[14]</sup> LTP 모듈과의 호환성 시험을 통해서 수행하였다.

ION은 LTP, BP를 비롯한 DTN 프로토콜 소프트웨어



(a) Test preparation

(b) Test results

그림 7. LTP 엔진 기능 시험 결과  
Fig. 7. Test result of LTP engine function.

웨어를 구현하여 오픈소스로 배포되고 있으며 tests 디렉토리에 각 모듈의 기능 시험 및 회귀 시험 (regression test)을 위한 테스트 절차 및 도구들을 제공하고 있다. 이 시험들은 기본적으로 BP의 기능을 시험하기 위한 것으로 LTP 계층만을 테스트하기 위한 절차는 포함하고 있지 않다. 하지만 LTP 세그멘테이션 데이터 송수신 시험을 위한 ltpdriver와 ltpcounter라는 도구를 제공하고 있기 때문에 이 도구들을 사용하고 기존 테스트 형상을 참조하여 LTP 호환성 시험을 위한 환경을 구성하였다.

먼저 ION tests 디렉토리 내에 ltp-retransmission 폴더를 복사하여 ltp-interoperability 등의 이름으로 저장하고 내부 시험 스크립트에 bpdriver(bundle sending test utility)와 bpcounter(bundle receiving test utility)를 호출하는 부분을 ltpdriver(LTP segmentation sending test utility)와 ltpcounter(LTP segmentation receiving test utility)를 수행하도록 수정하였다. ltp-retransmission 테스트는 DTN node 2와 DTN node 3 사이에 ltp retransmission 기능을 시험하기 위한 것으로, bpdriver와 bpcounter를 ltpdriver와

ltpcounter로 변경하여 DTN node 2와 DTN node 3 사이의 ltp 전송 기능을 시험할 수 있다. ltpdriver와 ltpcounter 구동을 위한 파라미터는 ION 기술 문서 (JPL D-48259)를 참고하여 적절히 기입하였다.

본 연구에서 구현한 LTP와의 호환성 시험을 위해서는 1개의 ltp 노드만 필요하므로 위 설정에서 LTP node 3에 대한 구동 부분만 남기고 LTP node 2에 대한 부분은 삭제하였다.

수신 시험을 위해서는 ION LTP node 3을 ltpdriver를 구동하도록 구성하고, 반대로 송신 시험 기능을 시험하기 위해서는 ION LTP node 3을 ltpcounter를 구동하도록 구성한다. 시험 도중 발생하는 LTP 노드간 트래픽을 와이어샤크(wireshark)로 녹화한 후 분석하여 LTP 표준에 맞도록 통신이 수행되었는지 확인하였다.

#### IV. LTP 활용 및 비전

우주 통신이 우주패킷프로토콜 기반의 점대점 통신에서 자동화된 우주 통신 네트워크로 진입하기 위해

서는 지상 인터넷의 TCP/IP 프로토콜과 같이 신뢰성 있는 점대점 통신 수단이 확보되어야 한다. 이러한 기술적 요구사항에 대해서 IETF(Internet Engineering Task Force, 국제 인터넷 표준화 기구)와 CCSDS를 비롯한 기술 표준화 단체에서 고민 끝에 내놓은 해결책이 바로 LTP이다. 그림 8에서 보여지는 바와 같이 LTP는 인접 노드와의 링크를 담당하며 CGR (Contact Graph Routing) 번들 라우팅 엔진을 보조하여 DTN 노드를 구성한다. 번들 처리 계층은 BSS(Bundle Streaming Service), CFDP(CCSDS File Delivery Protocol), AMS(Asynchronous Message Service), AMP(Asynchronous Management Protocol) 등의 응용 계층과 인터페이스 한다. LTP의 또다른 의미는 우주패킷프로토콜(SPP)을 보조한다는 것이다. SPP는 재전송 기능이 없기 때문에 미리 위성-지상국 간 접촉 계획(contact plan)을 세워 가장 신뢰성있게 데이터를 수신할 수 있는 경로(path)를 계획하여 데이터를 송수신한다. 만약 통신 도중에 복구할 수 없는 오류가 발생하면 그 오류가 비록 전체의 일부에 불과하더라도 전체를 버리는 방식으로 사용되었다.

이러한 단점을 극복하기 위해서 심우주파일전송프로토콜(CFDP)이 고안되어 사용되었는데 2009년 발사된 달정찰위성(Lunar Reconnaissance Orbiter, 미국 NASA LRO)에서 사용된 이래 미국 NASA 우주 탐

사 임무에 널리 사용되고 있다<sup>15-17</sup>. CFDP 프로토콜은 비신뢰성 전송 모드는 Class-1 모드와, 신뢰성 전송 모드인 Class-2, 그리고 중계 기능을 가진 Class-3, -4가 있는데 Class-2 모드는 각 노드간 CFDP만 사용될 때 적용되고, DTN 노드의 어플리케이션의 하나로 사용될 때는 Class-1 모드만 사용되는 식으로 정리되었다. DTN 개발이 본격화되면서 중계기능(Store and Forward, S&F)을 지원하는 Class-3 및 4 모드는 사실상 관리 중지(deprecated)되었다.

LTP 신뢰성 통신 방식을 SPP 기반 우주통신 노드 간 적용하게 되면 신뢰성있는 패킷 송수신 기능을 포함하여 CFDP Class-2를 사용하지 않고도 신뢰성있는 파일 전송이 가능하다. 신뢰성 전달 책임은 LTP와 같은 융합 계층이 담당하기 때문이다. 일례로 패킷을 파일로 저장하는 것은 단순하기 때문에 SPP+LTP 조합으로 CFDP Class-2 기능을 대체할 수 있을 것으로 판단된다. 라우팅 기능이 필요하지 않는 점대점 우주통신 통신에서 LTP 신뢰성 통신 기능을 사용하면 점대점 우주통신에서 DTN 기능을 사용하는 것과 동일한 이득을 얻을 수 있다.

점대점 노드 구성이 점점 확장되어 다수의 점대점 노드 구성이 포함되면 네트워크 단계로 확장되는 것이고 이때 번들 라우팅 로직이 필요하게 된다. 따라서 점대점 우주통신 환경에서는 SPP+LTP 조합이 최적의 형상이 될 수 있다. 자동 재전송 기능에 따라서 버려지는 패킷이 최소화되기 때문에 굿풋(goodput) 성능이 개선된다.

번들 프로토콜 표준이 BPv6에서 BPv7으로 개정되고 있고, 현재 LTPv1이 소프트웨어 플랫폼 상에서 구동하기에 최적화된 형상이라면 LTPv2는 하드웨어 플랫폼에서도 구동될 수 있도록 국제 관련 커뮤니티에서 개정이 논의 중이다. LTP 통신은 점대점 우주통신에서 우주통신 네트워크로 진입하기 위한 필요조건이라는 의미가 있다. 현재는 외국 우주 기관 및 우주 기업에서 상당 부분 기술 개발이 진척되고 성숙화되어 있기 때문에 DTN 기술 표준화 과정에서 BP 및 LTP 프로토콜의 개정에 대한 공통 의견을 모으는데 진통이 발생하고 있다. 향후 관련 기술력 확보와 국제적 우주통신 네트워크 자원을 활용하기 위해서 국내에서도 LTP 뿐 아니라 번들 프로토콜 및 라우팅 기술에 대한 분석 및 레퍼런스 구현에도 꾸준히 관심을 기울여야 할 것으로 생각된다.

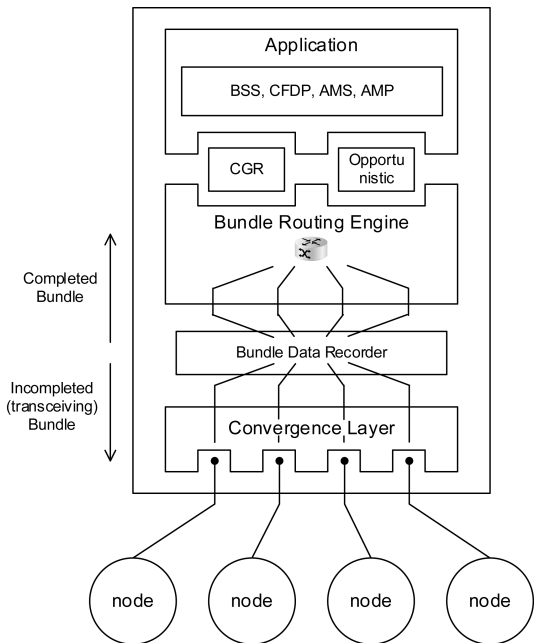


그림 8. DTN 시스템 내부 접속 개념도  
Fig. 8. Internal interface inside a DTN system.



## V. 결 론

new space 시대를 맞이하여 지구궤도 위성 임무뿐만 아니라 달, 화성을 포함한 심우주 탐사선 임무 개발 수요도 크게 증대될 것으로 예측된다. 이에 따라 점대점 우주통신 기반이 점차 우주통신 네트워크로 발전할 것으로 기대되며, 이에 따라 점진적인 우주통신 네트워크 기술을 구현하고 실험적 적용을 통해 그 가능성을 확인하고 기술 수준을 실제 상용 수준에 이르기까지 발전시켜 나가야 할 것이다.

본 논문에서는 중단허용네트워크 프로토콜의 융합 계층으로 사용되는 LTP 프로토콜을 자체 구현한 결과를 수록하였고 이를 활용하여 점대점 우주통신 및 우주통신 네트워크 구성에 적용하여 얻을 수 있는 효과와 그 방안을 고찰하였다.

LTP 프로토콜은 SPP 상에서 신뢰성 통신 수립의 타당하고 현실적인 방안을 제공하며, 이런 이유로 DTN 프로토콜의 융합계층을 담당하게 되었다. 네트워크 구성이 불필요한 점대점 우주통신 임무에서는 실질적으로 CFDP Class-2 모드를 대체할 수 있으며 우주에서 CFDP Class-1 모드 대신에 FTP(File Transfer Protocol)과 같은 대중적인 소프트웨어를 사용하는 것도 가능하리라 생각한다.

향후 우주 탐사가 더욱 활성화되면 DTN 기술을 통해 지상 인터넷과 동일한 수준의 데이터 송수신, SNS, 이메일, 화상회의 등의 서비스가 지원될 수 있을 것으로 기대되고 있다. 본 LTP 통신 기술은 점대점 우주통신 환경에서 SPP 단독 사용에 비해 많은 이점을 가지고 있음으로 통신 리소스가 매우 제한적인 달 표면 로버, 달 표면 과학 센서와 착륙선간 통신 등에 활용될 수 있을 것으로 생각된다.

본 연구가 차후 번들링 프로토콜 개발 및 라우팅 로직 개발 등 중단허용네트워킹 프로토콜 개발에 활용 가능할 것으로 기대한다.

## References

[1] wikipedia, [https://en.wikipedia.org/wiki/Origin\\_of\\_language](https://en.wikipedia.org/wiki/Origin_of_language)

[2] CCSDS, “*Space Packet Protocol - Recommendation for Space Data System Standards*,” CCSDS 133.0-B-1, Blue Book, Tech. Rep., 2003.

[3] W. Zhang, G. Yang, F. Jiang, K. Zhao, J. Wang, W. Li, and Z. Cheng, “Licklider

transmission protocol for GEO-relayed space internetworking,” *Wirel. Netw.*, vol. 25, no. 7, pp. 3747-3757, 2019.

- [4] J. B. Schoolcraft, S. C. Burleigh, R. M. Jones, E. J. Wyatt, and J. L. Torgerson, “The deep impact network experiments, concept, motivation and results,” in *Proc. SpaceOps 2010 Conf.*, 2010.
- [5] CCSDS, “*Solar System Internetwork (SSI) Architecture*,” CCSDS 730.1-G-1, Green Book, Info. Rep., 2014.
- [6] K. Scott and S. C. Burleigh, “*RFC 5050: Bundle Protocol Specification*,” IRTF DTN Research Group, 2007. [Online] Available: <https://tools.ietf.org/html/rfc5050>
- [7] M. Ramadas, S. C. Burleigh, and S. Farrell, “*RFC 5326, Licklider Transmission Protocol Specification*,” IRTF DTN Research Group, 2008. [Online] Available: <https://tools.ietf.org/html/rfc5326>
- [8] A. P. da Silva, S. Burleigh, and K. Obraczka, *Delay and Disruption Tolerant Networks: Interplanetary and Earth-Bound-Architecture, Protocols, and Applications*, CRC Press, pp. 213-215, Jul. 2018.
- [9] CCSDS, “*Licklider Transmission Protocol (LTP) for CCSDS*,” CCSDS 734.1-B-1, Blue Book, Tech. Rep., 2015.
- [10] D. Israel and D. Cornwell, “*LLCD-Disruption Tolerant Networking Demonstrations over LLCD's Optical Links*,” 2014, Retrieved Sep. 3, 2019 from <http://ipnsig.org/wp-content/uploads/2014/02/LLCD-DTN-Demonstration-IPNSIG-Final.pdf>
- [11] W. Eddy and E. Davies, “*RFC-6256: Using Self-Delimiting Numeric Values in Protocols*,” Internet Research Task Force, 2011. [Online] Available: <https://tools.ietf.org/html/rfc6256>
- [12] C. H. Koo, “Study for performance optimization for SDNV transformation function,” in *Proc. The Korean Soc. Aeronautical and Space Sci. Fall Conf.*, pp. 1171-1172, Nov. 2019.
- [13] [www.appsloveworld.com/download-sample-mp4-video-mp4-test-videos](http://www.appsloveworld.com/download-sample-mp4-video-mp4-test-videos)

- [14] NASA, “*Interplanetary Overlay Network(ION) Design and Operation*,” ver 3.7.2, JPL D-48259 Retrieved from <https://sourceforge.net/projects/ion-dtn/>
- [15] J. Wilmot, “*Use of CCSDS File Delivery Protocol (CFDP) in NASA/GSFC’s Flight Software Architecture: Core Flight Executive (CFE) and Core Flight System (cFS)*,” NASA GSFC, Code 582.
- [16] C. R. Tooley, M. B. Houghton, Jr R. S. Saylor, C. Peddie, D. F. Everett, C. L. Baker, and K. N. Safdie, “Lunar Reconnaissance Orbiter Mission and Spacecraft Design,” *Space Sci. Rev.*, vol. 150, no. 1, pp. 23-62, 2010.
- [17] CCSDS, “*CCSDS File Delivery Protocol (CFDP) - Draft Recommended Standard*,” CCSDS 727. 0-P-4.1, Pink BOOK, Tech. Rep., 2014.

구 철 회 (Cheol Hea Koo)



1997년 2월: 충남대학교 전자공학과 졸업

1999년 2월: 충남대학교 의용전자공학 석사

2021년 2월: 충남대학교 컴퓨터공학과 박사

2002년 3월~현재: 한국항공우주연구원 선임연구원

<관심분야> 내장형 소프트웨어, 위성 통신, 심우주 통신, CFDP, 우주인터넷

[ORCID:0000-0002-7180-1476]