

# 우주 인터넷 환경에서 AMP 기반의 원격 소프트웨어 업데이트 프레임워크

황호연\*, 조인희\*, 이경락\*\*

\*한양대학교 컴퓨터소프트웨어학과

\*\*한국전자통신연구원 무인이동체연구실

kd3122@hanyang.ac.kr, iwjoe@hanyang.ac.kr, krlee@etri.re.kr

## AMP-Based Remote Software Update Framework for Space Internet

Ho-Yeon Hwang\*, In-Whee Joe\*, Kyung-Rak Lee\*\*

\*Dept. of Computer Science, Han-Yang University

\*\*Unmanned Vehicle Research Section, ETRI

### 요약

우주 네트워크는 고지연성, 저신뢰성 등 기존의 지상 네트워크와는 차별화된 특성을 가진다. 또한 높은 예산이 투입되는 우주 임무 특성상 지상 시험 시 발견되지 않은 소프트웨어 버그가 막대한 비용을 초래할 수 있다. 그러나 현재까지 우주 네트워크 환경에 특화된 소프트웨어 업데이트 관련 연구는 부족한 실정이다. 따라서 본 연구에서는 Asynchronous Management Protocol(AMP)을 기반으로 원격 소프트웨어 업데이트 프레임워크를 새롭게 제안한다. 해당 프레임워크는 Interplanetary Overlay Network(ION) 소프트웨어를 사용하여 프로토타입을 구현하였으며, 실제 업데이트 케이스를 적용하여 기능 및 실사용성을 검증하였다.

### 1. 서론

최근 국제적으로 민간 주도의 우주 개발 산업이 급속도로 성장하고 있으나, 우주 환경에 적합한 원격 소프트웨어 업데이트 방법에 대한 연구는 극히 미비한 실정이다. 하지만 이는 성공적인 우주 임무를 수행하기 위해 반드시 필요한데, 그 이유는 수많은 인력과 막대한 예산이 투입되는 우주 임무의 특성상 지상 시험 시 발견되지 않은 소프트웨어 버그가 존재할 경우 예기치 못한 환경에서 기체의 결함을 초래하여 장기간 계획된 프로젝트가 성과 없이 중단될 수 있기 때문이다.

### 2. 관련 연구

우주 인터넷은 고지연성, 빈번한 링크 단절 등 기존의 지상 네트워크의 동작 환경과 구별되는 다양한 특징들을 지니고 있다. 예를 들어, Mars 2020 Perseverance 로버의 경우 화성과 지구 사이의 거리에 따라 편도 기준 최소 4분에서 최대 24분까지의 전송 지연이 발생한다.[1]

Delay Tolerant Network(DTN)는 이러한 특수한 통신 환경에 특화된 서비스를 제공하기 위해 고안된

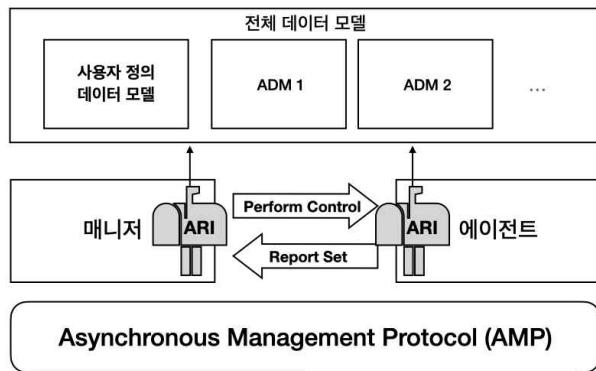
새로운 네트워크 패러다임이다. DTN은 Bundle Protocol(BP)[2]을 기반으로 한 계획적·확률적 라우팅, Store-and-forward 기법, Custody 노드를 통한 재전송 의무 전방 양도 등의 기능을 수행하며, 우주와 같은 극한의 상황에서도 신뢰성 있는 통신이 가능하도록 한다.[3]

Asynchronous Management Protocol(AMP)은 이러한 DTN 환경에서 사용하기 위한 목적으로 제안된 비동기 네트워크 관리 프로토콜이다. AMP는 지상망에서 사용되고 있는 기존의 네트워크 관리 프로토콜인 SNMP와는 달리 에이전트가 로컬에 구성된 State Based Rule(SBR) 등을 지속적으로 평가함으로써 자율적으로 각 프로토콜 관련 파라미터를 정의, 변경, 삭제한다.[4]

AMP를 통한 네트워크 관리의 한 주기는 주로 다음과 같은 방식으로 이루어진다. 먼저 매니저가 AMP Perform Control(PC) 메시지를 에이전트에게 보내면, 에이전트는 해당 PC 메시지의 내용을 수행한 후 필요시 AMP Report Set(RS) 메시지를 통해 수행 결과를 비동기적으로 매니저에게 보고한다.[5]

AMP는 Application Data Model(ADM) 및 Asynchronous Management Model Resource

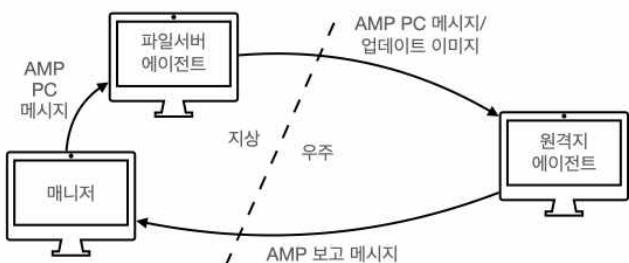
Identifier(ARI)를 통해 응용 기능별 항목들을 관리한다. ADM은 통계 데이터인 Externally Defined Data(EDD)와 수행할 동작이 정의된 Control(CTRL) 등의 데이터 모델들을 DTN 시스템 상에서 동작하는 프로토콜별로 구현한 것이며, ARI는 해당 데이터 모델들을 식별하기 위한 고유 식별자이다.[6] 이를 사용하면 사용자가 새로운 응용 기능에 대한 데이터 모델을 추가하는 것이 매우 용이하다. 그림 1은 AMP 및 ADM의 개념을 도식화한 것이다.



(그림 1) AMP 및 ADM 개념도

### 3. 원격 소프트웨어 업데이트 프레임워크

네트워크 아키텍처는 그림 2와 같이 지상 노드인 매니저 및 파일서버 에이전트와 우주에 존재하며 소프트웨어 업데이트의 대상이 되는 노드인 원격지 에이전트로 구성된다. 파일서버 에이전트는 그림 2에서는 매니저 노드와 물리적으로 다른 노드에 존재하고 있지만 매니저 노드에 같이 존재할 수 있다. 모든 참여 노드는 DTN 노드로서 DTN 통신을 한다.



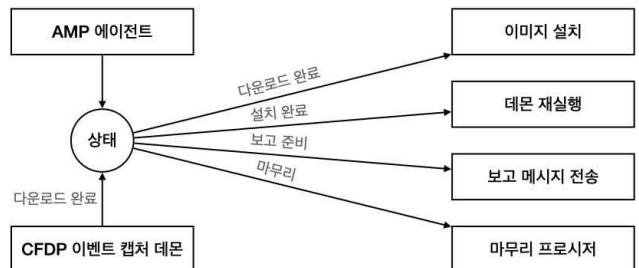
(그림 2) 네트워크 아키텍처

지상망은 TCP/IP 기반 표준 인터넷 프로토콜을 사용하여 통신한다. 매니저는 원격지 에이전트를 직접 관리하는 과학자의 개인 컴퓨터(PC)에 위치한다. 파일서버 에이전트는 Mission Operations Center(MOC) 내에 존재하거나 또는 허가 받은 운용자의 PC 내에 위치하고, 로컬 디스크의 정해진 위치에 업데이트 이미지가 저장되고 버전이 업데이트

된다. 마지막으로 원격지 에이전트 노드는 업데이트 이미지와 AMP PC 메시지를 받아 실제로 업데이트를 수행하는 노드를 의미하며, 탐사선, 착륙선, 궤도선, 로버 등 다양한 형태의 우주선이 이에 해당된다.

업데이트 순서는 다음과 같다. 먼저 매니저가 파일서버 에이전트로 AMP PC 메시지를 전송하면, 파일서버 에이전트는 원격지 에이전트로 로컬 파일 시스템에 저장된 최신 버전의 이미지를 CCSDS File Delivery Protocol(CFDP)[7]을 통해 전송하고, AMP PC 메시지를 바로 이어서 전송한다. 따라서 매니저는 단 한 번의 명령만을 전송하면 되므로 사용이 매우 간편하다. 뿐만 아니라, 파일서버 에이전트가 원격지 에이전트에게 업데이트 이미지와 AMP PC 메시지를 한 번에 전송하기 때문에 두 노드 간 제한된 통신 일정을 효율적으로 사용 및 관리할 수 있다.

원격지 에이전트에서는 준비, 다운로드 완료, 설치 완료, 보고 준비, 마무리 총 5개의 상태 정보를 관리하며, 보고 준비 및 마무리 상태의 경우 다른 상태와 공존할 수 있다. 이를 통해 업데이트 프로시저 도중 에러가 발생하더라도 성공적으로 완료함은 물론 매니저에게 보고도 수행할 수 있도록 한다. 그림 3은 원격지 에이전트가 수행하는 업데이트 프로시저를 요약한 것이다. 초기 상태는 항상 준비 상태이기 때문에 그림 3에는 따로 표시되어 있지 않다.



(그림 3) AMP SBR 및 상태 기반 업데이트 프로시저

업데이트 프로시저를 구체적으로 살펴보면, 먼저 CFDP 이벤트 캡처 데몬이 동작하면서 업데이트 이미지의 수신이 완료되는 즉시 상태를 준비 상태에서 다운로드 완료 상태로 업데이트한다. 이어서 수신된 AMP PC 메시지는 원격지 에이전트에 SBR을 구성하여 현재 상태가 다운로드 완료 상태인지를 지속적으로 평가하여, 만약 해당 조건이 참이면 다음 단계로 진행하여 이미지 설치 및 데몬 재실행 동작을 수행한다. 이때, 만약 에러가 발생하여 정상적으로 완료되지 않은 경우 상태 정보에 바로 보고 준비 상태를 추가한다.

데몬 재실행 동작까지 정상적으로 완료되었다면 원격지 에이전트는 상태를 보고 준비 상태로 변경하고, 최초에 구성된 SBR의 지속적인 평가에 의해 업데이트 수행 결과를 매니저에게 보고하는 동작을 수행한다. 만약 프로시저 수행 도중 에러가 발생하였다면 에러 코드를 전송하여 매니저 측 운용자가 모니터링 후 즉시 조치를 취할 수 있도록 한다.

보고 메시지 전송이 완료되면, 원격지 에이전트는 마지막 단계로 마무리 프로시저를 수행하여 동적 할당된 메모리를 정리한다. 보고 메시지 전송과 마무리 프로시저에서는 전체 프로시저가 성공적으로 완료된 경우에만 상태를 준비 상태로 되돌리고, 그렇지 않은 경우에는 에러가 발생하였던 최종 상태 그대로 유지한다. 이를 통해 유사시 바로 최종 상태로부터 이어서 업데이트를 수행할 수 있게끔 설계하여 원격지 컴퓨터의 자원 소모가 최대한 절약될 수 있도록 하였다.

#### 4. 구현 내용

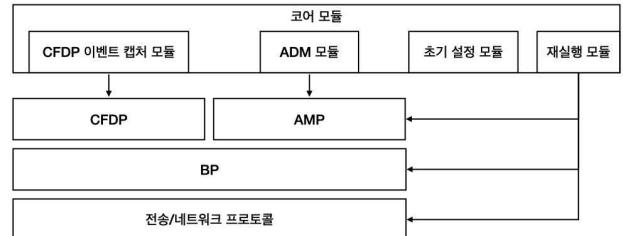
제안 프레임워크의 프로토타입을 구현하고 실사용성을 검증하기 위해 NASA의 JPL에서 우주 네트워크용 DTN 시스템을 구현한 Interplanetary Overlay Network(ION)라는 준 오픈 소스 소프트웨어를 사용하였다.[8] 개발 언어는 C를 사용하였다.

매니저와 파일서버 에이전트에 해당하는 구현 내용은 ADM 모듈 하나로만 구성하여 손쉽게 기존의 AMP 시스템에 통합될 수 있도록 하였다. 두 ADM 모듈은 파일서버 에이전트가 원격지 에이전트에게 업데이트 이미지와 AMP PC 메시지를 전송하는 동작을 수행하는 CTRL 오브젝트 및 이에 해당하는 ARI를 정의한 것으로, 프로그램 실행 시 초기화되어 메모리에 로드되도록 하였다.

원격지 에이전트의 경우는 ADM 모듈, 코어 모듈, 재실행 모듈, 초기 설정 모듈, 그리고 CFDP 이벤트 캡처 모듈까지 총 5개의 모듈로 나누어서 구현하였으며, 각 모듈의 구체적인 기능은 다음과 같다.

먼저 ADM 모듈은 업데이트 프로시저의 각 동작을 원격지 에이전트의 상태 정보와 연동하여 수행하기 위해, 동작은 CTRL 오브젝트로, 상태 정보는 EDD 오브젝트로, 현재 상태가 해당 상태인지를 판단하기 위한 조건은 SBR 오브젝트로 구현하였다. 즉 EDD를 통해 현재 상태를 읽고, SBR을 통해 현재 상태가 특정 상태일 경우에만 그에 매칭되는 CTRL을 수행하도록 하는 것이다.

나머지 모듈 중 코어 모듈은 설치 기능, 재실행 기능, 이미지 정보 관리 기능 및 기타 유ти리티 기능들을 포함하고 있으며 공유 라이브러리로 구현되었다. 재실행 모듈은 데몬 재실행 동작을 수행하는 단계에서 원격지 에이전트 프로그램 자체에 대한 재실행도 지원하기 위해 개별 실행 프로그램으로 빌드 하였으며 각 어플리케이션별로 종료 및 실행 기능을 구현하였다. 초기 설정 모듈은 원격지 에이전트가 실행될 때 이미 설치되어 있는 이미지들에 대한 정보를 메모리에 로드하여 추후 손쉽게 관리될 수 있도록 한다. 마지막으로 CFDP 이벤트 캡처 모듈은 파일서버 에이전트로부터 업데이트 이미지의 수신이 완료된 경우 상태 정보를 다운로드 완료 상태로 변경하는 기능을 수행하며, ION에 내장되어 있는 CFDP 어플리케이션을 부분적으로 수정하여 구현하였다. 전체 구현 내용을 정리하면 그림 4와 같다.



(그림 4) 구현 모듈 및 DTN 시스템과의 상호 작용

#### 5. 시험 결과

먼저 원격 소프트웨어 업데이트 프레임워크를 구현한 프로토타입이 실제 정상적으로 동작하는지 검증한 후, 간단한 시험 시나리오 구성을 통해 상태 정보를 기반으로 한 업데이트 프로시저가 정상적으로 수행되는지를 추가적으로 검증하였다.

그림 5는 4장에서 설명한 테스트베드를 사용하여 ION 플랫폼에서 구현한 프로토타입이 정상적으로 동작하는지를 검증한 결과 화면이다. 검증을 위한 업데이트 수행 이미지로는 ION에서 BP의 핵심 기능들이 구현되어 있는 BP 공유 라이브러리인 libbp.so와 데몬 프로세스인 bptransit을 선택하였다. 그림 5의 우측 사진을 보면 libbp.so 이미지 처리 시 이미 BP 관련 모든 데몬 프로세스들을 재실행하였기 때문에, bptransit 이미지 처리 시 재실행을 중복 수행하지 않는 것을 확인할 수 있다. bptransit의 경우 새로운 버전이 실행되었는지 검증할 목적으로 새로운 버전의 main 함수에 “hello from bptransit!”이라는 문구를 출력하는 코드를 추가하였는데, 실제로 재실행 시 화면에 잘 출력된 것을 확인할 수 있다.

```

----- AMP DATA REPORT -----
Sent to : 
Rpt Name : records
Timestamp : Tue Nov 10 01:04:47 2020
# Entries : 1
records : Install has been completed
Successfully restarted.

----- [root@ubuntu:/home/hoyeonjigil/AMP# [nm/age
nt/ingest.c:94] : rx_thread Group had 1 m
sgs
[nm/agent/ingest.c:95] : rx_thread Group
timestamp 0
[nm/agent/ingest.c:107] : rx_thread Recei
ved perform control msg.
***Download has been completed.
ctrl_start in>
        found image: libbp.so.
        found image: bptransit.
***Install has been completed.
ctrl_start ok<
        restart image: libbp.so.
        [Stop] bpclm,bpclock,bptransit,bp
uta,ipnadminep,ipnfw,lpclli,lpclco
        [Start] bpclm,bpclock,bptransit,bp
puta,ipnadminep,ipnfw,lpclli,lpclco
        restart image: bptransit.
        [Stop]
hello from bptransit!
***Restart has been completed.
ctrl_report in>
ctrl_report ok<
ctrl_fin in>
ctrl_fin ok<

```

(그림 5) AMP RS(보고) 메시지 내용(좌측)  
및 업데이트 수행 로그(우측)

또한 우측 사진의 `ctrl_start`, `ctrl_report`, `ctrl_fin`은 각각 3장에서 설명한 상태 기반 업데이트 프로시저 중 이미지 설치/데몬 재실행, 보고 메시지 전송, 마무리 프로시저 역할을 수행하는 CTRL 오브젝트이며, 순서대로 동작하고 있는 것을 확인할 수 있다.

그림 5의 좌측 사진은 보고 메시지 전송 동작이 수행된 후 매니저가 수신한 AMP RS 메시지를 출력한 것으로, 설치와 재실행이 정상적으로 완료되었다는 보고 메시지가 출력된 것을 확인하였다.

다음으로, 상태 정보를 기반으로 한 업데이트 프로시저가 정상적으로 수행되는지를 추가적으로 검증하기 위해 수신 업데이트 이미지 파일에 에러가 검출되어 이미지 설치 동작이 중단되는 시나리오를 모사하였다. 즉 이미지 설치 동작을 수행하는 함수가 압축 해제 도중 에러 코드를 반환하여 업데이트 프로시저가 중단되는 상황을 만든 뒤, 코드 복원 및 원격지 에이전트 재실행 후 마지막 상태로부터 업데이트 프로시저가 이어서 수행되는지를 검증하였다.

그림 6은 매니저에서 수신한 AMP RS 메시지를 출력한 것이다. 내용을 보면, 원격지 에이전트에서 이미지 설치 동작을 수행하는 함수인 `armurInstall` 함수가 비정상 종료되었으나, 추후 재실행 시 마지막으로 저장된 상태 정보인 다운로드 완료 상태로부터 시작하여 이미지 설치 및 데몬 재실행 동작까지 모두 정상적으로 수행되었고, 보고 메시지 전송 동작 역시 정상적으로 수행되었다는 것을 검증하였다.

## 6. 결론

본 논문에서는 우주 네트워크에 특화된 AMP의 SBR을 활용하여 최소한의 통신 일정만을 사용하여 원격지 노드에 대한 소프트웨어 업데이트를 효율적으로 수행할 수 있는 프레임워크를 제안하였으며, ION 소프트웨어를 통해 프로토타입 구현 및 실사용

```

----- AMP DATA REPORT -----
Sent to : 
Rpt Name : records
Timestamp : Tue Nov 10 00:48:34 2020
# Entries : 1
records : armurInstall failed., nm/utils/libarmur.c, 573
Install has been completed
Successfully restarted.

-----
```

(그림 6) 상태 기반 업데이트 프로시저 검증용  
AMP RS(보고) 메시지 내용

가능성을 검증하였다. 앞으로는 하위 버전과 호환이 되지 않는 주요 업데이트에도 적용이 될 수 있도록 프레임워크의 성능을 개선시키는 연구가 필요하다.

## 사사

본 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 한국연구재단(NRF)의 지원을 받아 수행된 연구임 [NRF-2016M1A3A9005564, 달 탐사용 우주 인터넷 기술개발]

## 참고문현

- [1] D. Baird “Space Communications: 7 Things You Need to Know” <https://www.nasa.gov/feature/goddard/2020/space-communications-7-things-you-need-to-know>, 2020.
- [2] K. Scott, S. Burleigh “Bundle Protocol Specification” RFC, IETF, RFC 5050, 2007.
- [3] K. Fall “A delay-tolerant network architecture for challenged internets” SIGCOMM '03, ACM, New York, NY, USA, 2003, 27-34.
- [4] E. Birrane “Asynchronous Management Architecture” (work in progress) Internet-Draft, IETF, draft-ietf-dtn-ama-01, 2021.
- [5] E. Birrane “Asynchronous Management Protocol” (work in progress) Internet-Draft, IETF, draft-birrane-dtn-amp-08, 2020.
- [6] E. Birrane, E. DiPietro, and D. Linko “AMA Application Data Model” (work in progress) Internet-Draft, IETF, draft-birrane-dtn-adm-03, 2018.
- [7] CCSDS File Delivery Protocol (CFDP), CCSDS 727.0-B-5, 2020.
- [8] S. Burleigh et al, ION-DTN, Jet Propulsion Laboratory(JPL), Pasadena, CA, USA, 2021.