

# 도커 컨테이너 기반 정보 중심 네트워크 구현

서동영, 강민욱, 정운원\*

숭실대학교

dyseo.tm@gmail.com, goodlookmw@gmail.com, \*ywchung@ssu.ac.kr(교신저자)

## Implementation of Docker Container-based Information Centric Network

Dong Yeong Seo, Min Wook Kang, Yun Won Chung\*

Soongsil University

### 요약

정보 중심 네트워크는 데이터의 이름을 식별자로 사용하여 통신을 수행하는 네트워크이다. 본 논문에서는 가상화 플랫폼인 도커-컨테이너 기술을 활용하여 이름 기반 라우팅을 수행하는 정보 중심 네트워크 기능을 설계 및 구현하였다.

### I. 서론

정보 중심 네트워크(ICN: Information-Centric Network)에서는 콘텐츠의 이름(Name) 정보를 사용하여 라우팅을 수행한다. 콘텐츠 데이터를 원하는 요청자는 콘텐츠의 이름을 포함한 Interest 패킷을 네트워크에 전송하고, 해당 콘텐츠의 생산자 혹은 캐싱 노드는 Interest를 수신하면 콘텐츠를 요청자에게 전송한다. ICN 라우터는 데이터 캐싱 기능인 CS(Content Store), Interest의 입출력 인터페이스 정보를 관리하는 PIT(Pending Interest Table) 및 이름정보 기반 포워딩 관리 테이블인 FIB(Forwarding Information Base)를 통해 라우팅을 수행한다[1, 2].

네트워크 기능 가상화(NFV: Network Function Virtualization)는 물리적인 네트워크 장비의 기능을 범용 서버에서 가상화하여 제공하는 기술을 말하며[3] 크게 VM(Virtual Machine) 기반 가상화 방식과 컨테이너 기반 가상화 방식으로 나눌 수 있다[4]. VM은 하이퍼바이저를 통한 하드웨어 가상화를 통해 독립적인 OS(Operating System)가 실행 가능한 구조로 다양한 OS 환경의 서비스를 하나의 물리 서버에서 제공할 수 있다는 장점이 있지만 게스트 OS에 필요한 커널 및 라이브러리 등으로 인한 이미지 자원이 크고 성능 저하가 발생할 수 있다. 반면 컨테이너 기술은 응용 프로그램을 실행을 위한 커널을 호스트 OS와 공유하면서 필요한 바이너리 및 라이브러리를 사용하기 때문에 하드웨어 자원 의존도가 낮으며 경량화된 방식으로 메모리 접근, 파일 시스템, 네트워크 속도, 환경 구성 측면에서 빠른 성능을 보이는 장점이 있다.

본 논문에서는 컨테이너 기반 오픈소스 가상화 플랫폼 엔진인 도커(Docker)를 사용하여 ICN 플랫폼을 설계 및 구현하고 가상화된 VNF(Virtualized Network Function)의 효율적인 관리 및 배포, 서비스 동작 등의 환경 구현을 위한 대화형 파일 시스템을 개발한다.

### II. 관련 연구

NDN(Named Data Networking)은 ICN의 대표적인 구조를 제공하는 프로젝트로 Interest 및 Data 패킷이 콘텐츠의 이름을 식별 정보로 활용하여 라우팅을 처리하며 CS, PIT, FIB 기능을 갖는 NDN 엔진을 통해 포워딩을 수행하는 메커니즘을 갖는다[5]. NDN 프로젝트에서는 그림 1과 같

이 NDN 기반 네트워크와 응용을 구현하고 테스트하기 위한 모듈을 제공하고 있다[6]. NFD(Named Data Networking Forwarding Daemon)는 NDN 플랫폼의 포워더로 동작하는 핵심 모듈로, Interest 및 Data 패킷을 통한 NDN 프로토콜 포워딩 동작을 위한 CS, PIT, FIB 기능을 포함하고 있다. NFD를 중심으로 각 컴포넌트(ndn-cxx, NDN-CLL) 간 상호작용을 통해 NDN 통신을 수행한다.

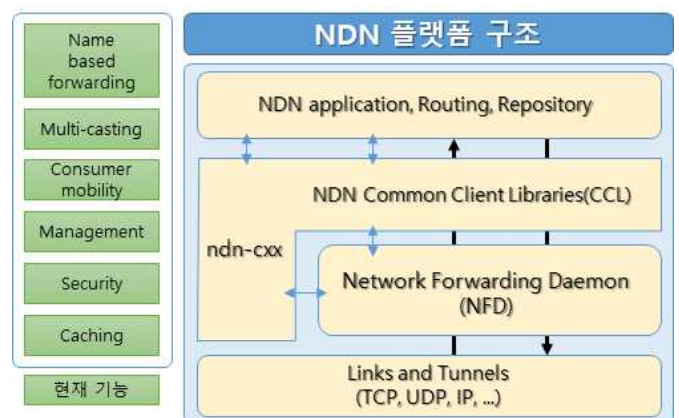


그림 1 NDN 플랫폼 구조 (NDN Platform Architecture)

리눅스 컨테이너(LXC, LinuX Container) 기반 가상화 기술인 도커는 VM 기반 가상화 기술보다 경량화된 서비스 개발 및 운영 환경을 지원한다. 도커-컨테이너는 리눅스 커널의 컨트롤 그룹(cgroup)을 통해 격리된 공간에서 자원을 제어하고 namespace 기능을 통해 자원을 분리하여 가상화된 독립 공간을 제공하며 chroot(change root) 기능을 통해 파일을 분리하여 독립된 공간을 생성 및 보안을 제공한다. 도커는 이러한 기능들을 통해 컨테이너라 불리는 독립된 공간 및 시스템 환경을 제공하고 프로세스 단위의 동작을 수행한다. 도커는 컨테이너 실행에 필요한 최소 단위의 프로그램, 라이브러리, 소스 등이 저장된 파일 형식의 도커 이미지를 사용하기 때문에 이미지 크기가 비교적 작아 빠른 설치, 프로그램 이식성, 쉬운 유지관리 등의 장점이 있다[7].

### III. 도커-컨테이너 기반 ICN 기능 구현

본 논문에서 제안하는 도커-컨테이너 기반 정보 중심 네트워킹 플랫폼 구조는 그림 2와 같다. 제안 구조에서는 유럽 표준화 기구인 ETSI에서 정의하는 NFV 참조 구조[8]를 기반으로 NFVI(NFV Infrastructure)를 설계하였다. Linux Ubuntu OS가 설치된 물리적 서버에서 도커 엔진을 사용하여 이름 기반 정보 중심 네트워크(NDN) 기능에 필요한 커널 모듈, 라이브러리 및 바이너리 자원을 Host OS와 공유할 수 있도록 하였다. 또한 NDN 프로젝트에서 제공하는 NDN 플랫폼(NFD, ndn-cxx)을 사용하여 컨테이너 VNF(C-VNF)를 생성하기 위한 도커 이미지 생성 과정을 담은 도커파일(Docker file) 형식을 정의하였다. 생성된 NDN 플랫폼 기반 도커 이미지를 통해 NFD 기능을 갖는 각 컨테이너들이 빠른 속도로 실행될 수 있으며 이러한 컨테이너들은 네트워크 노드가 되어 이름 기반 정보 중심 네트워크 기능을 수행할 수 있다.

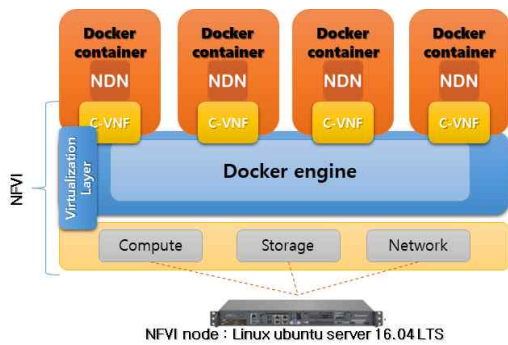


그림 2 도커-컨테이너 기반 NDN 플랫폼 구조

그림 3은 도커-컨테이너 기반 NDN 플랫폼의 동작 예시를 보여주고 있다. 현재 IP 기반 네트워크에 NDN 기능을 구현하기 위해 NDN 플랫폼에서는 TCP 혹은 UDP 페이스의 FIB 저장 기능을 사용하며 제안 구조에서는 컨테이너 환경에서 NDN 플랫폼의 기능이 수행되도록 설정하였다. 따라서 그림 3의 예시와 같이 정보 중심 네트워크의 노드 역할을 하는 각 컨테이너들은 PIT, FIB, CS 기능을 갖는 NFD를 사용하는 NDN 통신이 이뤄지도록 설정되었다. A/B/C라는 데이터를 생산자(Producer)가 게시하였을 때 같은 이름의 데이터를 요청하는 Interest 패킷이 발생할 경우, 각 노드(컨테이너)들은 FIB를 참조하여 해당 데이터의 이름을 기반으로 라우팅을 수행할 수 있도록 동작한다.

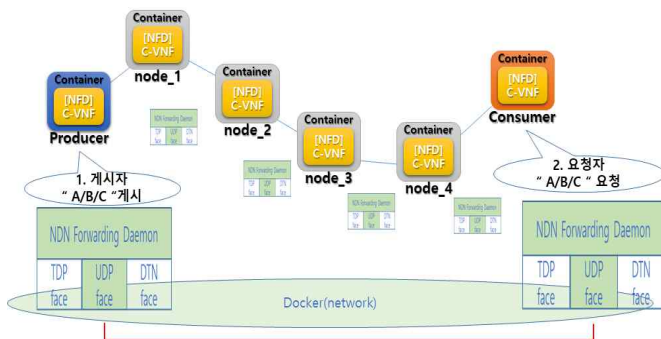


그림 3 도커-컨테이너 기반 NDN 플랫폼 동작

그림 4는 도커-컨테이너 기반 NDN 플랫폼의 실행 화면을 보여준다. 게시자 컨테이너 노드는 NDN 명령어를 통한 /A/B/C라는 데이터를 게시하였을 때, 요청자 컨테이너 노드는 해당 데이터를 요청하는 Interest 패

킷을 망에 전송한다. 중계 노드들은 /A/B/C라는 이름 PIT를 통해 해당 Interest가 게시자에게 전송될 수 있도록 하고 게시자는 역경로를 통해 해당 데이터를 요청자에게 전송하는 NDN 통신의 예를 보여주고 있다. NDN 기능을 갖는 컨테이너 생성 및 배포 시 요구되는 컨테이너 이름과 네트워크 주소 등의 설정 간편화를 위해 대화형 셸스크립트 파일시스템을 적용하였다. 파일 시스템은 C-VNF의 생성 시 명령창과의 입출력 대화 형식을 통한 이름 중복 확인, 네트워크 대역 확인 및 할당 설정을 가능하도록 하여 간편화된 컨테이너 생성 및 배포 절차를 적용하였다.

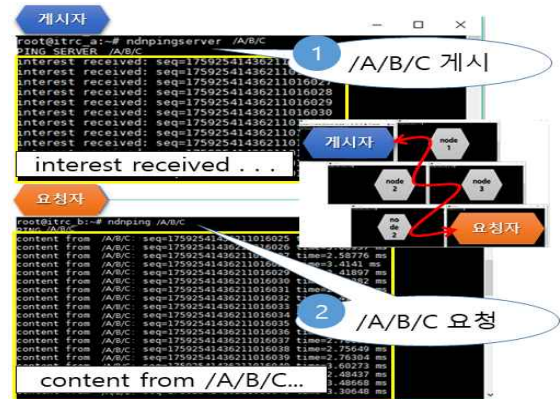


그림 4 도커-컨테이너 기반 NDN 플랫폼 실행 화면

### IV. 결론 및 추후 연구

본 연구에서는 컨테이너 기반 가상화 기술인 도커를 사용하여 컨테이너 기반 ICN을 설계 및 구현하였다. 추후 다수의 컨테이너 환경에서 효율적인 컨테이너 기반 정보 중심 네트워크 관리를 위해 컨테이너 오케스트레이션 기술과의 연동 방안을 연구하고자 한다.

### ACKNOWLEDGMENT

“본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음” (IITP-2020-2017-0-01633). 이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No.2017-0-00613, MEC 환경에서의 콘텐츠 기반 지연 감내 네트워킹 기술 개발).

### 참 고 문 헌

- [1] D. Meyer, L. Zhang, and K. Fall, “Report from the IAB Workshop on Routing and Addressing,” IETF RFC 4984, September 2007.
- [2] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros and G. C. Polyzos, “A Survey of Information-Centric Networking Research,” IEEE Communications Surveys & Tutorials, vol. 16, no. 2, pp. 1024-1049, July 2013.
- [3] ETSI ISG, “Network Functions Virtualization Introductory White Paper,” <http://portal.etsi.org/portal/server.pt/community/NFV/367>.
- [4] W. Felter, A. Ferreira, R. Rajamony and J. Rubio, “An updated performance comparison of virtual machines and Linux containers,” IEEE International Symposium on Performance Analysis of Systems and Software, Philadelphia, PA, pp. 171-172, 2015.
- [5] D. Saxena, V. Raychoudhury, N. Suri, C. Becker and J. Cao, “Named Data Networking: A survey,” Computer Science Review, Vol. 19, pp. 15-55, February 2016.
- [6] Named Data Networking(NDN) Project, <https://named-data.net/>
- [7] J. Fink, “Docker: a Software as a Service, Operating System-Level Virtualization Framework,” CodeLib Journal, vol. 25, 2014
- [8] ETSI GS NFV 002 v1.2.1, “Network Functions Virtualisation (NFV): Architectural Framework,” [https://www.etsi.org/deliver/etsi\\_gs/NFV/001\\_099/002/01.02.01\\_60/gs\\_NFV002v010201p.pdf](https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf)