Proceeding of KNOM 2018 Workshop

2018년 통신망운용관리워크숍 논문집

일시: 2018. 11. 30(금)

장소: SK Telecom 서울 본사, 4층 SUPEX Hall

주최: 한국통신학회 통신망운용관리연구회

후원: SK Telecom

한국통신학회 통신망운용관리연구회

초대의 말씀

한국통신학회 통신망운용관리 연구회 (KNOM)는 2018년 통신망 운용관리 워크샵 (KNOM Workshop 2018)를 통하여 통신망 운용관리 기술의 최신 연구 개발 현황을 국내 관련분야 학자, 연구원, 네트워크 관리자, 및 실무 담당자들에게 소개하고, 활발한 토론을 할 수 있는 장을 마련하고자 초청 강연 및 연구 논문 발표로 행사를 마련하였습니다.

클라우드 컴퓨팅, 모바일 컴퓨팅 등 향후 컴퓨터의 사용 개념을 혁신할 새로운 컴퓨팅 기술이 각광을 받고 있습니다. 또한 영상 데이터의 급속한 확산은 유무선을 포함한 모든 네트워크에서 상상하기 어려운 새로운 데이터 폭주로 이어지고 있으며 언제 어디서나 동영상을 볼 수 있는 컴퓨팅 체계가 마련되고 있습니다. 빠른 통신기술의 발전과 보급은 무선 Data Explosion이라는 새로운 문제를 야기하고 있고 이를 해결하는 것이 통신 사업자의 가장 큰 이슈가 되어, 지난 10여 년간 통신망 운용관리 분야의 주요 연구 주제였던 End-to-End 네트워크 관리는 유무선 통합 네트워크 환경에서 네트워크와 서버 및 단말을 포함해 관리해야 하는 현실적인 문제로 대두되었습니다. 또한 클라우드 컴퓨팅의 보편화는 네트워크 구조와 트래픽의 흐름을 근본적으로 바꾸어가고 있으며, 네트워크와 서비스에 대한 보안 침해도 급격히 증가하여 통신망 운용관리 분야의 연구와 개발 범위 또한 급속히 넓어지고 그 중요성이 더욱 강조되고 있습니다.

이러한 추세를 반영하여 KNOM Workshop 2018에서는 통신망 전반에 대한 모델링, 설계, 서비스 제공, 운용관리 및 보안 기술 분야의 최신 연구 개발 결과에 대한 유익한 정보제공과 토론의 장을 제공할 계획입니다. 부디 본 Workshop이 운용관리 전반에 걸친 활발한 기술교류와 토론의 장이 될 수 있도록 각 분야의 전문가 여러분들의 적극적인 논문투고와 발표를 기대합니다.

2018. 11.

한국통신학회 통신망운용관리연구회 위원장 최태상

2018 통신망운용관리워크숍 조직위원장 주홍택

운영위원회

조직위원장	주홍택(계명대)
학술위원장	석우진(KISTI)
학술위원회	정종문(연세대), 박수현(국민대), 박형곤(이화여대), 백상헌(고려대) , 석승준(경남대)
프로그램위원장	김진철(SKT)
프로그램위원회	박용석(삼성전자), 옥기상(KT), 이영석(충남대), 이재오(한국기술교육대), 조부승(KISTI)
등록 및 재정	최미정(강원대)
홍보	김윤희(숙명여대)
현장 및 진행	김명섭(고려대)
자문위원장	최태상(ETRI)
자문	송왕철(제주대), 김영탁(영남대), 홍충선(경희대), 이영우(KT) 최덕재(전남대), 홍원기(POSTECH), 유재형(POSTECH), 우왕돈(인소프트), 이경휴(ETRI)

프로그램

2018년 11월 30일(금)	
08:30-09:30	**등록 및 참석 안내**
09:30-10:45	**TS1. Blockchain** (5편 발표)
10:45-11:00	Coffee Break
11:00-12:15	**TS2: SDN/NFV** (5편 발표)
12:15-13:30	Lunch
13:30-14:30	**IS1: How close are we toward Autonomic Networking** 최태상 박사(ETRI)
13:30-15:30	**IS2: Opportunities and Challenges of Blockchain Development in SK Telecom** 정효진 모듈장(SKT)
15:30-15:50	Coffee Break
15:50-16:50	**IS3:Cyber-Physical System: 현황 및 주요 이슈** 박경준 교수(DGIST)
16:50~17:50	**IS4: 5G C-V2X 표준 및 기술현황** 김덕경 교수 (인하대)
18:00	KNOM OC Wrap-up Meeting

논문 목차

Technical Session 1. 블록체인(11월 30일(금) 9:30 - 10:45, SUPEX홀)
(좌장: 석승준 교수, 경남대)

1. 실시간 블록체인 네트워크 모니터링 시스템의 데이터베이스 시스템 설계
한도경, 방지원, 최미정 (강원대)

2. 이더리움 컨트랙트 모니터링 및 분석시스템
고경찬, 이채현, 홍원기 (포항공대)

3. 이더리움 기반 Hands-up & Go 분산 어플리케이션 개발
이기영, Hartanto Kurniawan, Intan Permatasari, Yoga Andrian, 주홍택(계명대)

4. Densification Power Law 기반 비트코인 네트워크의 통계 데이터 분석
백의준, 신무곤, Huru Hasanova, 김명섭(고려대)

5. Juggling Drones: Distributed Drone Port Approach to Public Drone Services
Jared Lynskey, Hong Choong Seon (경희대)

Technical Session 2. SDN/NFV(11월 30일(금) 11:00 - 12:15, SUPEX홀)
(좌장: 김윤희 교수, 숙명여대)

1. Service based Network Slice Selection Function
Javier Diaz Rivera, Talha Ahmed Khan, Mehmood Asif, Rafiq Adeel, Wang-Cheol SONG (제주대)

2. 가상 네트워크를 위한 OpenFlow 기반 가상 게이트웨이
이도영,김희곤,유재형,홍원기(포항공대)

3. SDN 기반 모바일 엣지 컴퓨팅 환경에서 머신러닝을 이용한 태스크 오프로딩 방안 연구
김기태,홍충선 (경희대)

4. SDN 기반의 OpenStack 네트워킹을 위한 Virtual TAP 설계 및 구현
정세연, 유재형, 홍원기(포항공대)

5. xDEditor: ETSI NFV Release-3 표준 준수 Network Service Descriptor 설계 및 관리 시스템 개발
이용승, 배병관, 김동을 (모비젠)

주최

KICS 한국통신학회

KNOM KICS 1996 통신망운용관리연구회 한국통신학회

후원

SK telecom

2018년 12월

일	월	화	수	목	금	토
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

실시간 블록체인 네트워크 모니터링 시스템의 데이터베이스 시스템 설계

한도경, 방지원, 최미정*
*강원대학교

*{dkhan, jiwonbang, mjchoi}@kangwon.ac.kr

Design of Database system of real-time blockchain network monitoring system

Dokyeong Han, Jiwon Bang, Mi-Jung Choi*
*Kangwon National Univ.

요 약

최근 비트코인의 시세 급변으로 인해 많은 사람들이 암호화폐에 관심을 가지며, 비트코인의 핵심 기술인 블록체인에 대한 관심도 높아졌다. 블록체인은 기존의 중앙집중형 방식인 은행과 달리 네트워크 참여자들 간의 신뢰를 기반으로 개인 거래가 이루어지는 분산형 방식이다. 블록체인의 무결성, 익명성 등의 특징을 악용한 불법 거래에 대한 문제가 존재한다. 암호화폐 불법 거래를 탐지하기 위해서는 블록체인 네트워크 모니터링 시스템이 필요하다. 본 논문에서는 블록체인 네트워크에서 데이터를 수집하여 악용사례와 불법 거래를 탐지하는 모니터링 시스템을 제안하고, 블록체인 네트워크 모니터링 시스템 중 수집한 데이터를 안정적으로 저장하는 데이터베이스 시스템을 설계한다. 수집한 데이터를 효율적으로 저장할 수 있는 데이터베이스 스키마와 일반적으로 많이 사용하는 데이터베이스들을 분석한 결과를 기반으로 실시간 블록체인 네트워크 모니터링에 적합한 데이터베이스를 제시한다.

Ⅰ. 서 론

최근 비트코인의 시세 급변으로 인해 많은 사람들이 암호화폐에 많은 관심을 가지게 되었으며, 비트코인의 핵심 기술인 블록체인 또한 지속적인 연구를 통해 재평가되어 잠재적인 기술 중 하나로 세계적으로 인정받고 있다. 비트코인은 사토시 나가모토라는 익명의 개발자가 2008년 10월 암호화 기술 커뮤니티에 게시한 “Bitcoin: A Peer-to-Peer Electric Cash System”[1]에서 처음으로 등장하였다. 은행과 같은 제3자의 개입 없이 개인 간 거래를 할 수 있는 전자화폐이다. 비트코인의 특징으로는 P2P(Peer-to-Peer) 네트워크의 문제점 중 하나인 서로 간의 신뢰 문제를 작업증명과 같은 합의 알고리즘을 통해 해결하여 무결성을 보장하며, 누구나 거래를 볼 수 있지만 거래자의 정보는 드러나지 않아 익명성을 가진다[2]. 이러한 블록체인의 특징을 활용해 금융, 보안, 네트워크, IoT(Internet of Things) 등 여러 분야에 블록체인을 적용한다면 많은 발전이 이루어질 것으로 예상되고 있다[3]. 반대로 익명성을 악용하여 마약거래, 사기, 불법 무기 거래, 돈세탁 등의 범죄에 암호화폐를 사용하는 사례가 발생하고 있다[4]. 암호화폐가 불법적으로 사용되는 것을 방지하기 위해서 블록체인 네트워크를 모니터링하는 시스템과 불법거래를 탐지하는 기법이 필요하다. 본 논문에서는 블록체인 네트워크 모니터링을 위한 전체 시스템을 제안하고, 수집한 데이터를 안정적으로 저장하기 위한 데이터베이스 시스템을 설계한다. 또한 블록체인 네트워크 모니터링 시스템의 요구를 충족시킬 수 있는 데이터베이스를 조사하고 시스템에 적합한 데이터베이스를 제시한다.

Ⅱ. 관련연구

블록체인 익스플로러는 비트코인, 이더리움[5] 등 암호화폐에 대한 통계정보를 웹 기반 어플리케이션으로 만드는 프로젝트이다. 블록체인 익스플로러는 블록체인 네트워크에 접속하지 않아도 블록체인 정보를 볼 수 있으며 대표적으로 Blockexplorer, Etherscan, Blcokchain.info 등이 있다. 하지만, 블록체인 익스플로러는 불법거래에 대한 판별이 불가능하다. 이를 극복하기 위해 블록체인 네트워크 모니터링에 대한 연구가 필요하다. 대표적인 블록체인 네트워크 모니터링 관련 연구인 BitIodine[6]은 비트코인 네트워크 내에서 사용자의 경로 및 역 경로를 추적하는 기능을 제공한다. Elliptic과 Chainalysis도 비트코인을 사용한 자금 세탁 및 사이버 범죄자를 추적하는 기능을 제공한다. 본 논문에서는 블록체인 네트워크 모니터링 시스템에 대한 설계를 제시하고, 그 중 수집한 데이터를 저장하는 데이터베이스 시스템을 설계하였다.

Ⅲ. 실시간 블록체인 네트워크 모니터링 시스템

본 논문에서 참고한 실시간 블록체인 네트워크 모니터링 시스템[7]은 수집 에이전트와 노드 인터페이스, 데이터베이스 시스템 데이터 분석 엔진, 시각화를 위한 웹 서버로 이루어져 있다. 그림 1은 실시간 블록체인 네트워크 모니터링 시스템의 전반적인 구조이다. 수집

에이전트를 통해 각 블록체인 네트워크마다 수집된 데이터는 노드 인터페이스를 통해 모니터링 서버로 전송된다. 수집한 데이터를 데이터베이스에 저장한 후 분석 엔진을 통해 불법거래 탐지, 보안 공격 탐지, 블록체인 포렌식을 한다. 마지막으로 웹 서버에서는 수집한 데이터와 탐지 및 분석 데이터를 시각화하여 UI로 제공해 주는 기능을 담당한다.

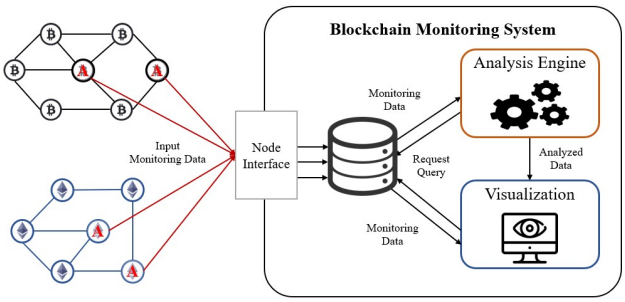


그림 1. 실시간 블록체인 네트워크 모니터링 시스템

실시간 블록체인 네트워크 모니터링 시스템은 노드 인터페이스를 통해 블록체인 네트워크에 대한 데이터가 실시간으로 입력된다. 이러한 데이터를 안정적으로 저장하기 위한 데이터베이스 시스템 구조를 설계하였다. 그림 2는 실시간으로 들어오는 데이터를 저장하는 데이터베이스 시스템의 구조를 나타낸 것이다. 먼저 모니터링 서버의 과부하를 막기 위해 실시간으로 전송된 데이터는 Apache Kafka로 입력된다. Apache Kafka는 실시간 로그 처리에 특화된 메시징 큐 시스템으로서 블록체인 네트워크 및 데이터 종류별로 Topic을 지정하여 입력된 데이터를 저장한다. 그 다음 Apache Storm을 사용하여 Apache Kafka에 저장된 데이터에서 분석에 필요한 데이터를 추출한다. 데이터베이스의 부하를 막기 위해 추출한 데이터를 다시 Apache Kafka를 거쳐 데이터베이스에 저장함으로써 블록체인 모니터링 시스템의 안정성을 높일 수 있다.

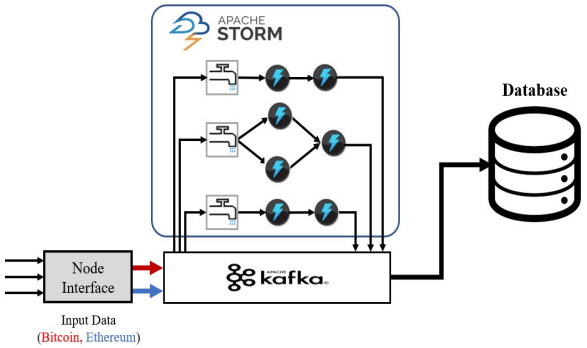


그림 2. 안정적인 저장을 위한 데이터베이스 시스템

그림 2의 데이터베이스 시스템을 통해 가공된 데이터를 저장하는 데이터베이스 구성 테이블을 제안한다. 블록체인 네트워크에서 수집할 수 있는 데이터들이 다양하고 대량의 데이터가 입력되기 때문에 단일 데이터베이스에서 관리하기가 까다롭고 병목현상이 발생할 수 있다. 이를 해결하기 위하여 블록체인의 구성요소를 구분하고 특징에 따라 분산 데이터베이스에서 안정적으로 데이터를 관리할 수 있도록 데이터베이스 스키마를 설계하였다. 그림 3은 블록체인 네트워크 모니터링 데이터베이스 스키마를 나타낸다. 블록 해시 값, 이전 블록 해시 값, 블록 사이즈 등을 저장한 블록 테이블, 계정 주소와 장부에 대한 정보를 저장하는 노드 테이블, 트랜잭션의 ID, From/To 정보, Input/Output Value 등 트랜잭션에 대한 정보를 저장하는 트랜잭션 테이블, BandWidth, throughput, 송신지 주소와 목적지주소, 프로토콜 등을 저장하는 네트워크 테이블, 컨트랙트 ID, 노드 ID, Timestamp 등을 저장하는 컨트랙트 테이블로 구성하였다.

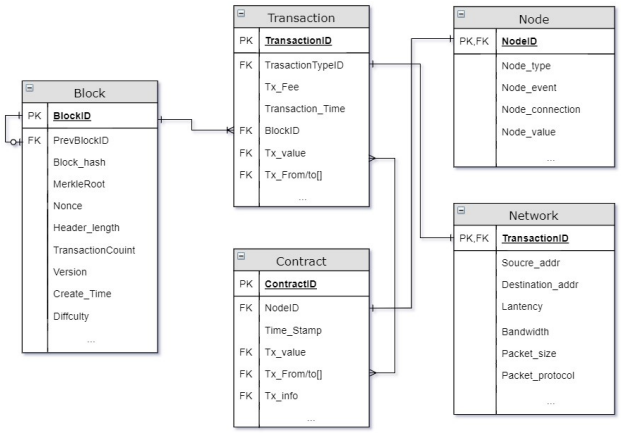


그림 3. 블록체인 모니터링 시스템 데이터베이스 스키마

기존 데이터베이스 성능 분석 결과를 비교해 실시간 블록체인 네트워크 모니터링에 적합한 데이터베이스를 제시한다. 앞에서 제안한 실시간 블록체인 네트워크 모니터링 시스템의 기능은 실시간성과 대량 저장이 요구된다. 데이터베이스의 종류 중 비 관계형 데이터베이스[8]는 빠른 읽기 및 쓰기, 대량 저장 지원, 저렴한 비용 등의 장점이 있어 블록체인 네트워크 모니터링 시스템의 요구를 충족할 수 있다. 블록체인 모니터링 시스템에 적합한 비 관계형 데이터베이스 중 오픈소스 기반 실시간 처리 시스템에 일반 대중적으로 사용되는 시스템들에 대한 성능을 조사하였다. RavenDB와 CouchDB는 읽기, 쓰기 및 삭제에 대한 작업속도가 상대적으로 저조하고, Cassandra는 읽기 작업은 느리지만 쓰기 작업과 삭제 작업에 적합하다. Couchbase와 MongoDB는 읽기, 쓰기, 및 삭제 작업에서 가장 빠른 결과를 보이지만, Couchbase는 모든 키 가져오기 기능을 지원하지 않는다[9]. 대표적인 비 관계형 데이터베이스 성능적인 측면을 고려하여 비교한 결과, 설계한 블록체인 네트워크 모니터링 시스템에 적합한 데이터베이스로 MongoDB로 결정하였다.

Ⅳ. 결론

블록체인의 무결성과 보안성, 탈중앙화와 같은 특성을 활용한 관련 연구가 금융, 의료, 전력 등 다양한 분야에서 활발하게 진행되고 있다. 하지만 블록체인의 특성을 악용하여 불법 거래, 범죄, 돈 세탁 등의 문제도 발생하고 있다. 본 논문에서는 블록체인 네트워크를 모니터링하여 수집한 데이터를 통해 비정상적인 거래나 불법 거래 등을 탐지 및 추적하는 모니터링 시스템을 제안하였고, 블록체인 네트워크 모니터링 시스템 중에서 수집한 데이터를 안정적으로 저장하기 위해 Apache Kafka와 Apache Storm을 통해 가공된 데이터를 효율적으로 저장하는 분산 데이터베이스와 데이터베이스 스키마를 설계하였다. 또한 일반적으로 많이 사용되고 있는 비 관계형 데이터베이스 중 성능 분석 자료를 통해 실시간 블록체인 네트워크 모니터링에 적합한 분산 데이터베이스를 제시하였다. 향후 연구로는 각종 악용행위와 불법거래에 대한 분석 및 탐지를 수행하는 실시간 블록체인 네트워크 모니터링 시스템 시스템을 구현할 것이다.

ACKNOWLEDGMENT

이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 ([No.2018-0-00539, 블록체인의 트랜잭션 모니터링 및 분석 기술개발], ['SW중심 대학(강원대학교)'])

참 고 문 헌

[1] Satoshi N. "Bitcoin: A peer-to-peer electronic cash system.", 2008.

[2] Pilkington M. "11 Blockchain technology: principles and applications," in *Proc. of* Research handbook on digital transformations 225, 2016.

[3] Foroglou G., Anna L. Tsilidou, "Further applications of the blockchain," May. 2015.

[4] Liao K., Zhao Z., et al. "Behind closed doors: measurement and analysis of CryptoLocker ransoms in Bitcoin," in *Proc. of* Electronic Crime Research (eCrime), 2016 APWG Symposium on. IEEE, Toronto, ON, Canada, Jun. 2016.

[5] Wood G. "Ethereum: A secure decentralised generalised transaction ledger," Ethereum project yellow paper 151, pp. 1-32. ,2014.

[6] Spangnulol M., Maggi D., et al. "BitIodine: Extracting intelligence from the Bitcoin Network," in *Proc. of* International Conference on Financial Cryptography and Data Security. pp. 457-468 Springer, Berlin, Heidelberg, Nov. 2014.

[7] Kyungchan K. et al. "Design of Monitoring and Analysis system on Blockchain network,", 2018

[8] Cattell, R. "Scalable SQL and NoSQL data stores," Acm Sigmod Record 39.4, pp. 12-27, Dec. 2011

[9] Li Y. and Manoharan S. "A performance comparison of SQL and NoSQL databases," in *Proc. of* Communications, computers and signal processing (PACRIM) on IEEE, Victoria, BC, Canada, Aug. 2013.

이더리움 컨트랙트 모니터링 및 분석시스템

고경찬 O, 이채현, 홍원기

포항공과대학교 컴퓨터공학과

{kkc90, chlee0211, jwkhong}@postech.ac.kr

Design of Monitoring and Analysis system on Contract in Ethereum

Kyungchan Ko O, ChaeHyeon Lee, James Won-Ki Hong

Department of Computer Science and Engineering, POSTECH

요 약

블록체인은 네트워크 참여자 모두에게 저장되는 정보에 대한 투명성을 공개하는데, 이것을 기반으로 하는 신뢰할 수 있는 공개된 분산 원장기술이다. 비트코인뿐만 아니라 스마트 컨트랙트 기능이 추가된 2 세대 블록체인 플랫폼 이더리움도 상당한 인기를 누리고있다. 이더리움도 블록체인에 저장되는 정보들을 모두 공개하며, 이로서 악의적인 사용자가 데이터를 의도적으로 수정하려는 시도를 네트워크 참여자 모두가 주시할 수 있다. 하지만 스마트 컨트랙트 정보도 함께 공유가되기 때문에 공개된 취약한 코드를 악용하여 해킹하려는 시도가 증가하고 있다. 이러한 해킹을 방지하기 위해서는 블록체인에 전파되어 저장되는 컨트랙트 관련정보들을 수집하여 컨트랙트의 취약성을 분석해야 한다. 본 논문에서는 컨트랙트 모니터링 및 취약한 컨트랙트를 탐지하는 방법에 대해서 소개하고 향후 연구 방향에 관해 기술하고자 한다.

I. 서 론

비트코인 [1]의 출현으로 비트코인의 기반 기술인 블록체인이 사람들 사이에 많이 언급되고 있다. 비트코인은 이전에 존재하는 여러 기술들을 집약하여 구현했지만, 이를 통해 블록체인기술을 세상에 알려고 사람들은 비트코인 이후로 블록체인을 언급했다. 그래서 비트코인은 1 세대 블록체인 기술이라고 알려져 있다. 비트코인 이후에 블록체인으로 구현한 많은 암호화폐들이 개발되기 시작했다. 그 중에서 스마트 컨트랙트[2]의 개념을 최초로 도입한 이더리움 [3]이라는 블록체인 플랫폼이 있다. 스마트 컨트랙트는 1994 년에 닉 사제보(Nick Szabo)가 최초로 제안한, 신뢰할 수 없는 컴퓨터 인터넷 환경에서 고도로 발달된 계약을 준수하도록 하는 프로토콜이다. 이더리움은 스마트 컨트랙트의 개념을 도입해서 블록체인 컴퓨팅 플랫폼으로서 구현되으며, 최초로 플랫폼의 역할을 할 수 있는 암호화폐 플랫폼의 출현이다. 이후로 많은 암호화폐 플랫폼들이 스마트 컨트랙트의 개념을 도입하려는 시도를 하고 있고, 그래서 이더리움을 2 세대 블록체인 기술이라고 알려져 있다.

최근에 블록체인 업계로 많은 자본이 투입되고 있는데, 이를 뒷받침 해주는 것이 이더리움의 스마트 컨트랙트이다. 이더리움에서는 스마트 컨트랙트를 이용해서 누구나 토큰(Token)을 발행할 수 있다. 이를 통해서 사람들은 ICO(Initial Coin Offering)을 진행하여 투자를 받는데, 이러한 부분에서 많은 투자금액이 유입되었다. 하지만 이러한 장점 뒤에는 스마트 컨트랙트의 취약점을 악용하여 해킹할 수 있다는 단점이 있다. 대표적인 해킹사례로 SMT(SmartMech) 토큰 무한 생성 해킹, 패리티 멀티시그 지갑 해킹(Parity Mulisig Wallet Hacked), DAO(Decentralized autonomous organization) 해킹 등이 있다. 이러한 해킹들은 이더리움 전체 네트워크에 악영향을 미치고 있다. 이러한 해킹사례들을 방지하기 위해서는 취약한 컨트랙트들을 탐지할 수 있는 모니터링 및 분석기법이 먼저 연구되어야 한다. 그리하여 본 여구에서는 컨트랙트 모니터링 방법과 이를 통해 수집된 정보를 이용해서 취약한 컨트랙트를 탐지하는 방법을 소개한다.

II. 관련 연구

1. Oyente

Oyente [4]는 싱가폴 국립대학에서 주관하는 연구 프로젝트의 일부로서, Symbolic execution 을 사용하여 이더리움 스마트 컨트랙트를 분석하는 도구이다. Oyente 는 오픈소스로 공개되었고 bytecode 레벨에서 분석 가능하기 때문에 Solidity, LLL, Serpernt, Viper 를 포함하는 모든 High-level EVM 언어들에서도 작동하기 때문에 범용적으로 사용될 수 있다.

2. Mythril

Mytril [5] 은 오픈소스로 공개되었고 이더리움 스마트 컨트랙트를 위한 보안 분석 도구이다. 이

도구는 스마트 컨트랙트의 다양한 보안 취약점들을 탐지하기 위해서 concolic analysis, taint analysis , control flow checking 등의 분석기법을 사용한다. 또한, Mytril 은 스마트 컨트랙트를 분석할 때 분석 대상으로 Solidity code, Solidity bytecode, Contract address 를 선택할 수 있다.

III. 컨트랙트 모니터링 및 분석시스템

컨트랙트 모니터링 및 분석시스템의 아키텍처는 전반적으로 그림 1 과 같은 구조를 갖는다. 추가적으로, 해당 아키텍처는 스마트 컨트랙트를 이용하는 대표적인 블록체인 플랫폼인 이더리움을 기반으로 설계되었다. 컨트랙트와 관련된 정보들로는 컨트랙트를 생성하는 트랜잭션, 컨트랙트 어카운트, 컨트랙트의 함수를 실행시키기 위한 트랜잭션 등이 있다. 따라서 정보들은 트랜잭션과 어카운트 정보를 모니터링 함으로써 얻을 수 있다. 이더리움 클라이언트 geth 를 구동시켜, Monitor 가 RPC 를 통해서 트랜잭션, 어카운트 정보를 모니터링 한다. Monitor 가 수집한 정보는 Preprocessor 에게 전달되어 전처리를 거친 후에 Database 에 컨트랙트 관련정보들이 저장된다. 두 개의 분석기(코드 기반, 통계 기반)는 이 정보들을 이용해서 트랜잭션의 취약성 정도를 판단하여 취약한 컨트랙트 주소를 추출한다. 추출된 정보는 Database 에 저장되고, 웹 서비스를 통해서 취약한 컨트랙트 주소 List 를 공개한다.

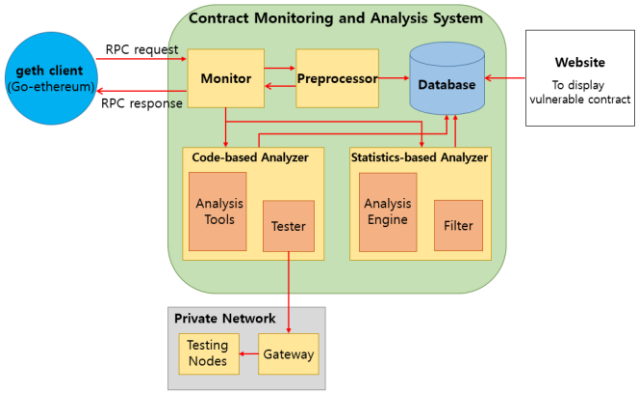


그림 1. 컨트랙트 모니터링 및 분석시스템

1. 코드 기반 분석기

코드 기반 분석기는 모니터링되는 트랜잭션들 중에서 'to' 필드가 nil(없음)인 트랜잭션들에 포함된 'data' 필드 내용을 이용해서 bytecode 레벨에서 컨트랙트를 검증한다. Analysis Tools 정적분석을 수행하며 Oyente, Mythril 등 bytecode 레벨에서 스마트 컨트랙트를 분석할 수 있는 도구들을 포함한다. 이것은 추후에 성능이나 정확도가 높은 분석도구가 개발되면 추가할 수 있도록 확장 가능한 구조로 설계된다. 정적분석뿐만이 아니라 Tester 는 추출된 bytecode 를 Private Network 의 Gateway 로 전달하여 해당 Gateway 에서 동일한 바이트 코드를 이용해서 컨트랙트를 생성하고, Private Network 상에서 여러 노드들이 일련의 패턴을 이용하여 동적분석을 실행한다. 정적 및 동적분석을 통과하지 못한 컨트랙트는 취약한 컨트랙트로 판단되기 때문에, Database 에 해당 컨트랙트 어카운트 주소를 전달한다.

2. 통계 기반 분석기

통계 기반 분석기는 지금까지 수집된 Historical 트랜잭션 및 컨트랙트 정보를 기반으로 분석한다. Analysis Engine 에서는 Historical 정보들을 수많은 DApp 및 스마트 컨트랙트 해킹사례들을 통해서 트랜잭션의 Outliner 를 추출한다. 예를 들어, 어느 하나의 외부 소유 계정에서 취약한 컨트랙트로 일정 패턴이나 한번에 많은 컨트랙트 함수 호출을 위한 트랜잭션을 보냈다면, 이를 이용하여 일정 주기마다 혹은 단시간에 일정 개수의 트랜잭션을 하나의 오부 소유 계정에서 생성하여 전파했다는 취약한 컨트랙트를 악용하려는 시도로 탐지할 수 있다. Filter 에서는 이러한 패턴이나 통계정보를 기반으로한 Outlier 에 걸리는 트랜잭션들을 추출하여 해당하는 외부 소유 어카운트와 취약한 컨트랙트 어카운트의 주소를 Database 로 전달한다.

IV. 결론 및 향후 연구

컨트랙트 모니터링 및 분석시스템은 취약점을 가지고 있는 스마트 컨트랙트가 해커들에게 해킹되어 악용되거나 일반적인 사용자가 잘못 사용되는 것을 막기 위한 핵심기술이다. 본 연구는 스마트 컨트랙트와 관련된 정보를 블록체인 네트워크에서 어떻게 수집할 수 있고, 수집한 데이터를 이용하여 취약점을 분석할 방법들에 대해 논의하였다. 또한, 이더리움 블록체인 플랫폼의 스마트 컨트랙트를 모니터링하고 분석하는 것에 초점을 맞추었다. 향후 본 연구에서 제안한 전반적인 구조를 실제로 구현하고 취약한 스마트 컨트랙트를 탐지하기 위한 다양한 연구를 진행할 수 있도록 할 예정이다.

ACKNOWLEDGMENT

이 논문은 2018 년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2018-0-00539)

참 고 문 헌

[1] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008).
[2] Buterin, Vitalik. "A next-generation smart contract and decentralized application platform." white paper (2014).
[3] Szabo, Nick. "The idea of smart contracts." Nick Szabo's Papers and Concise Tutorials 6 (1997).
[4] Oyente github: https://github.com/melonproject/oyente
[5] Mythril github: https://github.com/ConsenSys/mythril-classic

이더리움 기반 Hands-up & Go 분산 어플리케이션 개발

이기영, Hartanto Kurniawan, Intan Permatasari, Yoga Andrian, 주홍택

계명대학교

lkydig@naver.com, tanyatantociputat@gmail.com, intanishere16@gmail.com, yovie1703@gmail.com, juht@kmu.ac.kr

Development of Etherium based Hands up & Go Distributed Application

Kiyoung Lee, Hartanto Kurniawan, Intan Permatasari, Yoga Andrian, Hongtaek Ju

Computer Engneering, Keimyung University

요 약

블록체인은 참여자의 익명성, 거래의 투명성 그리고 기록의 불가역성이 특징이다. 이러한 특징은 여행이나 취미 활동을 위한 행사 진행에 적용하면 효과적이다. 참여자의 익명성은 참여자 모집에 도움이 되고 참여금 모금이나 행사 비용 사용이 투명하게 공개되며 행사 진행에 대한 완전한 기록이 보증된다. 본 논문에서는 이러한 장점을 가진 Hands Up & Go 블록체인 어플케이션을 이더리움기반으로 개발한 결과를 제시한다.

Ⅰ. 서 론

요즘에는 인터넷에서나 오프라인에서 단체가 소규모 행사를 진행하기 위하여 참가자를 모집하는 경우 많이 있다. 행사 진행에서 소요되는 비용과 수입 등을 고려하여 일정 이상의 참여자가 모집되어야 행사가 진행되는 경우도 있다. 이런 행사에 참석자의 익명성이 보장되면 행사나 프로그램에 참여할 때 보다 자유롭게 참여를 결정할 수 있으며 행사 진행 주관자는 많은 참여자 확보가 가능하다. 행사 참여자 모집할 때 참여자는 자신의 이름 등 정보가 공개되어 참여를 꺼리는 경우가 있기 때문이다. 대표적인 예시로는 동내 주민 투표, 단체 물건 구입 그리고 단체 운동, 여행 프로그램 참가자 모집 등이 있다. 참가자 모집부터 행사 종료까지 진행 과정이 참여자들에게 공개되지 않아서 투명성이 확보되지 않고 따라서 행사가 활성화 되지 않고 있다. 일정 이상의 참여자가 확보되어 행사가 진행되는 경우에 참여금액 모금, 행사 비용 지출 등 행사 진행 기록이 참여자에게 공개되어 투명하게 되면 행사가 활성화 된다. 또한 행사 진행과정에 대한 기록이 한번 기록되면 변경되지 않는 기록의 불가역성이 확보되면 행사가 원만하게 진행된다.

블록체인 기술은 참여자의 익명성, 기록의 투명성과 불가역성을 보장할 수 있는 기술이다.[1] 블록체인 기술 중에서 이더리움의 스마트 컨트랙트는 단순한 거래 뿐만 아니라 일련의 사건이나 거래 내용이 논리적인 흐름으로 진행되는 계약을 기록하고 진행하는 기술이다.[2] 이더리움 스마트 컨트랙트는 블록체인의 불변성을 보장하며 코드 특성상 지갑 주소만으로도 사용자를 구별 할 수 있기 때문에 익명성도 보장된다. Hands up & Go는 불특정 다수를 대상으로 특정 이벤트에 참여하는 익명의 참가자를 모집하는 이더리움 스마트 컨트랙트 분산 어플리케이션이다.[3] 본 논문에서는 익명성과 이벤트 진행 상황에 대한 투명성과 불가역성을 보장하기 위해서 위에서 설명한 이더리움 분산 어플리케이션 개발결과를 제시한다.[4] 논문의 구성은 다음과 같다. 2장에서 Hands Up & Go의 개념을 설명하고 3장에서 개발 할 때 고려한 사항에 대하여 논의하고 스마트 컨트랙트 설계 결과를 제시한다. 4장에서는 구현과 적용 결과를 검증하며 5장은 결과와 향후 연구 내용이다.

Ⅱ. Hands Up & Go 소개

Hands Up & Go은 행사 진행을 위한 블록체인 기반 분산 어플리케이션이다. 행사는 행사 내용 확정, 행사 공지, 참여자 모집, 행사 진행, 행사 마무리 순서로 진행된다. 행사 내용은 행사 주관자가 확정하며 행사 내용에는 행사의 이름, 간단한 설명, 일시, 장소, 최소 참여자, 최대 참여자 숫자, 참여금액, 참여자 모집 기한으로 이루어져 있다. 행사 주관자는 행사 내용이 확정되면 이를 공지한다. 공지한 내용은 행사 참여에 관심이 있는 모든 사람들에게 공지된다. 행사 참여에 관심이 있는 사람은 행사 참여를 선언하고 참여금액을 지불한다. 참여자자가 증가하여 참여 모집 기한 전에 최소 참여자 숫자를 초과하게 되면 행사는 공지 상태에서 진행가능 상태로 전환되고 참여자 모집 기한이 도래하거나 최대 참여자 숫자에 도달하면 모집완료 상태가 된다. 진행가능 상태나 모집완료 상태는 참여자에게 통지된다. 당연히 참여자 모집 기한에 도달하고 참여자가 최소 참여자 숫자를 만족시키지 못하면 행사는 행사 종료가 된다. 모집이 완료되면 행사가 진행되며 행사 주관자는 행사 시작과 행사 종료를 기록으로 남긴다.[5] 행사 시작과 행사 종료는 이를 증빙할 수 있는 증거자료로 남기게 된다.[6]

이 전체 과정이 블록체인 기반에서 실행된다. 행사 주관자와 사용자는 익명으로 분산 어플리케이션에 익명으로 가입하여야 하며 블록체인 네트워크를 통해서 연결이 되어 있다.[7] 행사 공지, 참여자 모집, 행사 진행, 행사 마무리의 각 단계는 블록체인 네트워크에서 정보를 공유하고 기록된다. 행사 주관자와 행사 참여자는 익명으로 진행되므로 개인 정보 누출이 되지 않기 때문에 많은 참여자를 모집할 수 있다. 행사 주관자가 불순한 의도를 가지고 불법적인 행사를 주관할 수 있는 위험성이 있다. 이러한 위험성은 공개 블록체인에서 실행되는 경우이며 사설 블록체인에서는 문제가 되지 않는다(사설에서는 익명성은 유지됨).[8] 모든 기록은 블록체인에 쌓이게 되고 누구나 알 수 있기 때문에 투명하며 행사 진행 기록이 보증된다.[9]

Ⅲ. 이론적 고찰 및 설계

블록체인 분산 어플리케이션(DApp: Decentralized Application)을 개발할 때 고려할 수 있는 블록체인 플랫폼으로 이더리움, NEO(.NET platform), Cardano, ICON 등이 있다. 이더리움은 최초의 DApp 개발 플랫폼이며 가장 널리 사용되고 있으며 NEO는 마이크로소프트의 .NET 플랫폼과 통합되어 있어 개발 환경이 잘 갖추어져 있고 이더리움의 중국 버전이라 불리는 Cardano와 한국에서 많이 사용하는 ICON 은 파이선 기반으로 만들어져 친숙해 질 수 있다는 장점이 있다. 본 연구에서는 관련 문서의 풍부하고 예제 및 개발 도구가 잘 갖추어진 이더리움 사용을 선택했다.[10]

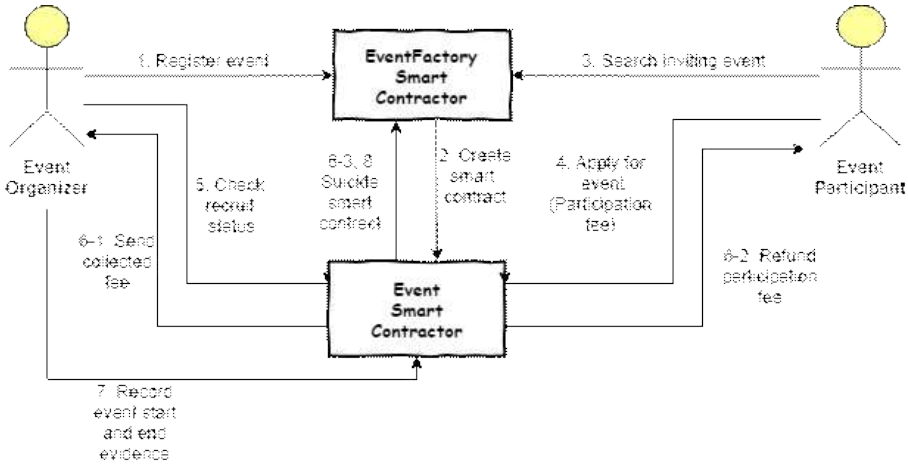


그림 1. Hands up & Go 스마트 컨트랙트 설계

새로운 블록체인 개발에 있어서 중요하게 다루어져야 할 내용 중에 하나가 합의 방법(Consensus Algorithm)이다. 작업 증명(PoW: Proof of Work), 지분 증명(Proof of Stake) 그리고 권한 증명(PoW: Proof of Authority)가 있다. 본 연구에서 채택한 합의 방법은 PoA이다. PoA는 권한을 가진 핵심 노드가 트랜잭션에 대한 검증을 책임지며 검증의 결과를 각 노드에 확산시키는 합의 방법이다. 이 방법은 PoW과 비교하여 합의 과정이 빠르고 마이닝에 코인이 소요되지 않으며 이더리움 공개 블록체인에서 주로 사용되는 PoS과 비교하여 합의 과정이 단순하고 사설 블록체인에 적용하기에 적합하다. 본 연구에서는 현재 개념 정립과 프로토타입 개발 단계이므로 PoA를 채택하였다.

블록체인 어플리케이션 개발에서 중요한 점 중에 하나가 토큰의 의미를 확정하는 것이다. 비트코인에서는 디지털 화폐의 의미로 규정할 수 있으며 다른 응용에서는 물질적인 자산의 소유에 대한 권리의 의미로 규정될 수 있다. 본 연구에서는 토큰을 화폐이기는 하나 행사 참여금 지급으로만 사용하는 화폐로 한정한다. 토큰의 생성 및 일반 통화나 타 화폐로의 변환에 대하여는 사업적인측면에서 다루어져야 할 내용이므로 본 연구서는 논의 밖의 주제이다. 합의 방법으로 PoA를 선택했으므로 마이닝에 관련된 토큰의 의미는 본 연구에서는 아직 논의할 단계는 아니다.

스마트 컨트랙트를 설계할 때 고려해야 할 것으로 토큰 지급과 환급 방법, 스마트 컨트랙트의 소유와 권한의 관계, 그리고 스마트 컨트랙트의 생성과 소멸에 대한 라이프사이클 관리이다. 스마트 컨트랙트에 오류가 발생한 경우의 대처나 보안에 관한 것도 주요 고려사항이나 본 연구에서는 아직 고려하지 않았다.[11]

그림 1은 Hands Up & Go의 스마트 컨트랙트 설계 구조이며 복수개의 스마트 컨트랙트로 고안되었다. EventFactory 스마트 컨트랙트는 Event 자식 컨트랙트를 생성하고 자식 컨트랙트의 주소와 상태를 저장한다. 행사 정보나 자식 컨트랙트 정보를 블록체인이 아닌 어플케이션에 저장할 수도 있으나 안정성과 투명성을 높이기 위해서 블록체인에 저장한다. 이 방법은 블록의 크기가 커지는 단점이 있다. Event 컨트랙트는 행사마다 하나씩 생성되는 스마트 컨트랙트이다. 이 컨트랙트는 유한 상태기계(State Machine)로 동작한다. 참가비는 행사 참가를 신청할 때 참가자가 지불하며 행사 개최 조건이 만족되어 개최되면 행사 주관자에게 보내지며 행사 개최 조건에 도달하지 못하면 행사 참가자에게 환불된다. EventFactory 스마트 컨트랙트는 소유의 개념이 무의하다. 누구나 행사를 등록하고 공지할 수 있으며 누구나 공지된 행사에 참여 신청을 할 수 있기 때문이다. 그러나 Event 스마트 컨트랙트는 행사 주관자 소유이어야 한다. 소유자만이 행사 진행 단계를 제어할 수 있으며 수집된 참가비를 수

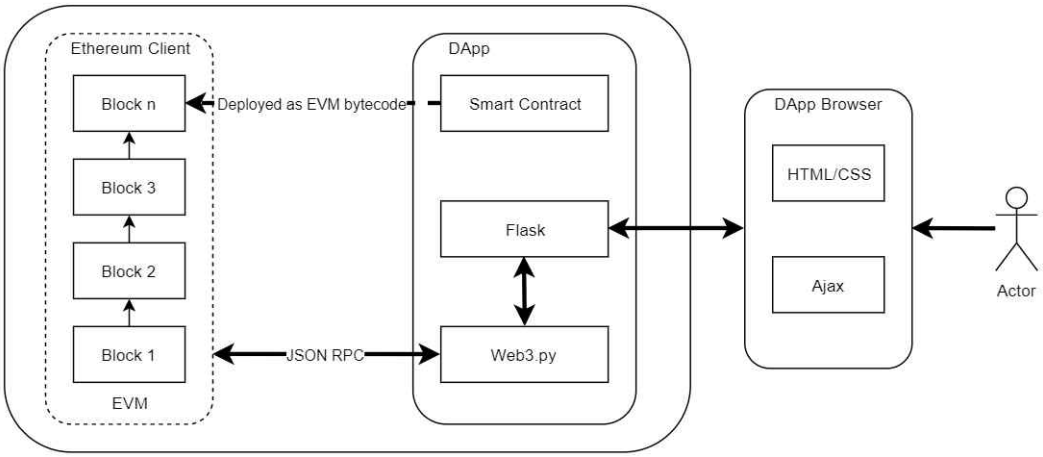


그림 2 Hands Up & Go 분산 어플리케이션 구조

령할 수 있기 때문이다.

Ⅳ. 구현 및 개발 결과

Hands Up & Go 분산 어플리케이션의 구현 구조는 그림 2와 같다. 이더리움 블록체인을 기반으로 블록이 생성되며 스마트 컨트랙트가 실행된다. Web3.py는 블록체인 네트워크와 RPC로 통신을 하며 Flask를 사용하여 소형 웹 서버를 구축하였다.[12] 노드에는 이더리움 Full Node와 이더리움 클라이언트 그리고 분산 어플리케이션 서버가 동작한다. 사용자는 웹 브라우져로 서버에 접속하여 사용한다. 웹 브라우져에서는 Ajax를 사용하여 비동기적으로 행사의 진행 상황 등이 갱신된다.

개발된 Hands Up & Go 분산 어플리케이션을 4개의 노드로 구성된 사설 네트워크에서 실행하였다. 구축된 실행환경에서 점심 모임 행사에 적용하였다. 점심 모임 행사에 점심 시간 전에 4명의 참가자가 모이면 각자 20 이더 참가금으로 특정 식당에서 점심 모임 행사를 시작하며 증빙으로는 영수증 사진을 등록하도록 하였다.

위의 행사에 적용하여 실행한 결과 생성된 블록은 총 7 이며 등록시 블록의 크기는 4206byte이고 추가적인 행위의 경우 538byte였다. 마이닝에 소요된 평균 시간은 19초이었다. 개발된 결과에 대하여 가장 직접적인 평가는 타당성 평가이나 본 논문에서는 효과적인측면, 사용성측면 등은 고려사항에서 벗어난 것이다.

Ⅴ. 결론 및 향후 연구

본 논문에서는 이더리움 블록체인 기반으로 행사 진행에 사용하는 분산 어플리케이션인 Hands Up & Go 개발 내용과 결과를 제시하였다. Hands Up & Go가 무엇인지 설명하고 이를 위한 스마트 컨트랙트 설계할 때 고려한 여러 가지 사항에 대하여 논의하였으며 설계 결과도 제시하였다. 스마트 컨트랙트를 이용한 분산 어플리케이션 구현 방법 및 실행 환경을 제시하였고 실제 특정 행사에 적용하여 실행한 결과도 제시하였다.

향후 연구로는 스마트 컨트랙트를 보안적인측면에서 검토하여 개선할 필요가 있으며 현재 개발된 스마트 컨트랙트는 모든 데이터가 블록에 저장되는 방법이나 어플리케이션 프로그램에 일부 데이터를 분산하여 성능을 개선하는 방안이 검토되어야 한다.

ACKNOWLEDGMENT

이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2018-0-00539, 블록체인의 트랜잭션 모니터링 및 분석 기술개발)

참 고 문 헌

[1] Maximilian Wohrer and Uwe Zdun. “Smart Contracts: Security Patterns in the Ethereum Ecosystem and Solidity”. 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE). IEEE. 2018.

[2] G. Wood. “Ethereum: A secure decentralised generalised transaction ledger”. Ethereum Project Yellow Paper, vol.151. 2014.

[3] [Online]. “solidity 0.4.18 documention”.https://solidity.readthedocs.io/en/v0.4.25/. 2018.

[4] Joris Bontje. “DApp Design Patterns”. https://www.slideshare.net/mids106/dapp-design-patterns. 2015, November.

[5] Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. “Hawk: The blockchain model of cryptography and privacy-preserving smart contracts”. In 2016 IEEE symposium on security and privacy (SP) (pp. 839-858). IEEE. 2016, May.

[6] Watanabe, H., Fujimura, S., Nakadaira, A., Miyazaki, Y., Akutsu, A., & Kishigami, J. “Blockchain contract: Securing a blockchain applied to smart contracts”. In Consumer Electronics (ICCE), 2016 IEEE International Conference on (pp. 467-468). IEEE. 2016, January.

[7] 김철진. “신뢰성 향상을 위한 이더리움 블록체인 기반의 온라인 투표 시스템”. 한국산학기술학회 논문지, 19(4), 563-570. 2018.

[8] 김세아, 원예종, 이지은, 최병주. “블록체인 기반 전자투표 시스템 설계 및 구현”. 한국정보과학회 학술발표논문집, 1931-1933. 2018.

[9] Yavuz, E., Koc, A. K., Cabuk, U. C., & Dalkılıc, G. “Towards secure e-voting using ethereum blockchain”. In Digital Forensic and Security (ISDFS), 2018 6th International Symposium on (pp. 1-7). IEEE. 2018, March.

[10] [Online]. “go-ethereum”. https://github.com/ethereum/go-ethereum. 2018

[11] Luu, L., Chu, D. H., Olickel, H., Saxena, P., & Hobor, A. “Making smart contracts smarter”. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 254-269). ACM. 2016, October.

[12] [Online]. “web3.py documention”. https://web3py.readthedocs.io/en/stable/. 2018

Densification Power Law 기반 비트코인 네트워크 통계 데이터 분석

백의준, 신무곤, 지세현, Huru Hasanova, 김명섭
고려대학교
{ pb1069, tm0309, sxzer, hhuru, tmskim }@korea.ac.kr

The Analysis of Bitcoin Network Statistical Data Based on Densification Power Law

Ui-Jun Baek, Mu-Gon Shin, Se-Hyun Jee, Huru Hasanova, Myung-Sup Kim
Korea Univ.

요 약

사토시 나카모토에 의해 블록체인 기술이 개발되고 비트코인이 새로운 암호화폐 시장을 개척한 이후 여러 암호화폐들이 등장하고 그 수와 규모는 나날이 증가하고 있다. 또한 블록체인 기술의 익명성과 여러 취약점을 이용한 범죄들이 발생하고 있으며 이에 취약점 개선과 범죄 예방을 위한 많은 연구들이 진행되고 있으나 범죄를 저지르는 사용자들을 탐지해내기엔 역부족이다. 따라서 네트워크 내 자금 세탁, 자금 탈취 등 이상 행위를 탐지 하는 것은 매우 중요하며 이에 본 논문에서는 비트코인 네트워크의 트랜잭션 및 User 그래프의 Feature들을 분석하고 네트워크 내 이상 탐지에 적절한 Feature들을 제시한다.

Ⅰ. 서 론

사토시 나카모토에 의해 블록체인 기술이 개발되고 비트코인이 새로운 암호화폐 시장을 개척한 이후 여러 암호화폐들이 등장하였으며 그 수와 규모는 나날이 증가하고 있다. 블록체인 시장의 급격한 성장에 따라 블록체인 기술의 익명성과 취약점을 이용하는 여러 범죄들이 발생하고 있으며 현재까지도 지속적으로 발생하고 있다. 취약점 개선과 범죄 예방을 위한 많은 연구들이 진행되고 있으나 악성행위를 예방하고 그 행위를 저지르는 악성 사용자들을 정확히 탐지해내기엔 역부족이다. 그러므로 네트워크 내 자금 세탁 및 탈취와 같은 이상 행위를 탐지하는 것은 매우 중요하며 본 논문에서는 비트코인 네트워크의 트랜잭션 및 User 그래프의 Feature들을 분석하고 이상 행위 탐지에 적절한 Feature들을 제시한다.

본 논문은 1장 서론, 2장 관련 연구, 3장 본론, 4장 분석 결과 순으로 설명하고 마지막 5장에서 결론과 향후연구를 제시한다.

Ⅱ. 관련 연구

[1]은 시간에 따른 그래프 변화의 특성을 설명한다. 정상적인 네트워크의 그래프의 노드와 에지 수가 로그 스케일 상에서 선형함수의 형태를 가진다는 Densification Power Law를 제시하는데 이 법칙에 따라 특정 네트워크 의 그래프의 분포가 비선형적일 경우 네트워크 내 이상이 있을 수 있다고 판단할 수 있다.

[2,3]은 블록체인 네트워크로부터 User 데이터를 추출하고 이를 특정기준에 따라 분류하거나 클러스터링 알고리즘을 통해 관련된 여러 지갑들의 연관성을 추출하는 방법을 제안하였다. [2,3] 모두 블록체인 네트워크를 분석하고자 하는 사용자에게 Forensic 분석의 가능성을 제시할 순 있으나 Heuristic한 기준과 수동적인 분석으로 시시각각 변화하는 네트워크의 특성을 모두 반영하기 힘들다는 한계점을 지닌다.

[4]는 비트코인 네트워크로부터 트랜잭션 데이터로부터 User 그래프와 트랜잭션 그래프를 추출하고 군집화하고 각 클러스터 내 이상치를 계산하는 수식을 통해 의심스러운 트랜잭션 혹은 User를 탐지하는 방법을 제안하였다. 그러나 탐지에 사용한 Feature들의 종류가 적어 정확한 탐지가 어렵다는 한계점을 지니며 이에 본 논문에서는 다양한 Feature과 그 통계정보를 Densification Power Law에 따라 분석하고 그 분석 결과를 통해 정확한 이상탐지에 있어 적절한 Feature들을 제시한다.

Ⅲ. 본론

본 장에서는 핵심 개념 및 데이터 수집 및 처리에 대해 설명한다.

i. Power Degree & Densification Power Laws

Densification Power Law는 노드(N)와 에지(E)로 이루어진 그래프에서 특정 시간 t의 노드 개수의 α제곱은 특정시간 t의 에지의 개수에 비례한다는 법칙이며 이는 수식 1과 같다.

$$E(t) \propto N(t)^{a} \quad (1)$$

이를 변형하여, 실제 정상적인 네트워크에서 $P(k)$를 차수 k를 가지는 노드의 Feature라고 정의하고 γ가 양의 정수일 때 $P(k)$는 차수 k의 역수에 비례하며 이는 수식 2와 같다. $P(k)$는 잔액, 트랜잭션 사이즈, 총 거래금액 등으로 대체될 수 있다.

$$P(k) \propto k^{-\gamma} \quad (2)$$

ii. 데이터 수집 및 통계 추출

비트코인 네트워크의 1부터 20만번 째 블록에 담긴 트랜잭션 데이터를 수집하였으며 트랜잭션 데이터로부터 User 데이터를 추출하였다. 이를 노드가 User 데이터인 노드 그래프, 노드가 트랜잭션인 트랜잭션 그래프의 형태로 변형하며 두 그래프는 입력과 출력이 존재하므로 방향성을 가지는 차수(In-Degree, Out-Degree)를 가진다. 마지막으로 그래프로부터 합, 최댓값, 최솟값, 평균, 표준편차를 추출하였으며 추출한 모든 Feature는 그림 [1,2]에 나타나 있다.

그림 1과 같이 User그래프에서 In/Out-Degree의 Number of Degrees 총 2개와 In/Out-Degree 2가지 방향성의 Value/Size/Weight 3개의 Features, 5개의 통계정보 총 30(2*3*5)개를 추출하였다.

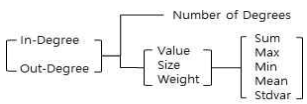


그림 1. User그래프에서 추출한 Feature Set

이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원(No.2018-0-00539-001,블록체인의 트랜잭션 모니터링 및 분석 기술개발)을 받고 2018년도 정부(교육부)의 재원으로 한국연구재단의 지원(NRF-2018R1D1A1B07045742)을 받아 수행된 연구임

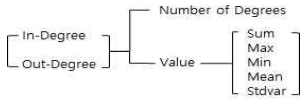


그림 2. 트랜잭션 그래프에서 추출한 Feature Set

그림 2와 같이 트랜잭션 그래프에서 In/Out-Degree의 Number of Degrees 총 2개와 In/Out-Degree 2가지 방향성의 Value, 이에 대한 5개의 통계정보 총 10(2*1*5)개를 추출하였다.

iii. 데이터 그래프 화

수집하고 추출한 데이터를 비교가 용이하도록 로그 스케일 그래프로 나타내었으며 x축에는 공통적으로 차수(In-Degree, Out-Degree)로 설정하고 y축은 추출한 각각의 Feature로 설정하였다.

Ⅳ. 실험 결과

본 장에서는 추출한 데이터를 그래프로 나타내고 이에 대해 설명한다.

서론에서 언급했듯이 그래프의 분포가 비선형일 경우 해당 네트워크 내이상이 있을 수 있다. 따라서 그래프의 분포를 보며 분석하고 비선형적인그래프의 분포를 가지는 Feature를 찾는다.

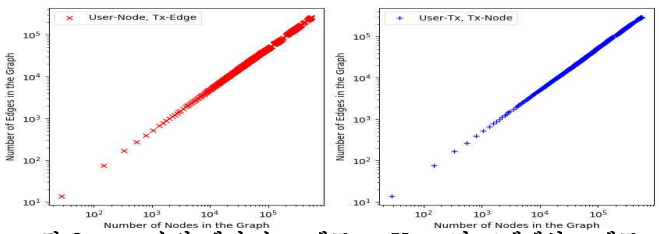


그림 3. 노드 수와 에지 수 그래프 - User 및 트랜잭션 그래프

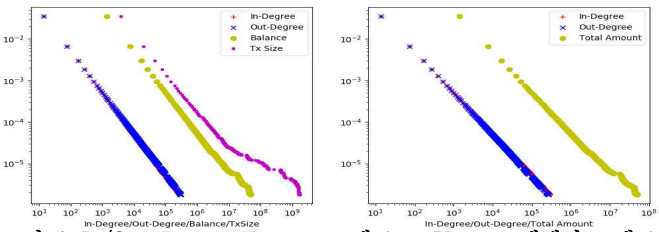


그림 4. In/Out-Degree Feature 그래프 - User, 트랜잭션 그래프

그림 3의 User 및 트랜잭션의 노드-에지 그래프와 [4]에서도 제시했던 그림 4의 In/Out-Degree에서는 단 한 개 Tx Size를 제외하고 모든 그래프가 선형함수의 형태를 나타내는 것을 확인했으며 이러한 Feature들로는 정상과 이상을 구분할 명백한 특징이 없다고 말할 수 있으며 정확한 이상 탐지가 어렵다고 판단된다.

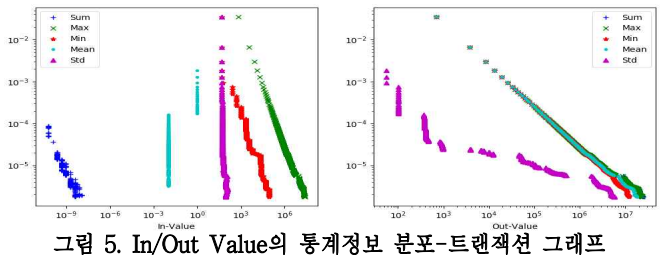


그림 5. In/Out Value의 통계정보 분포-트랜잭션 그래프

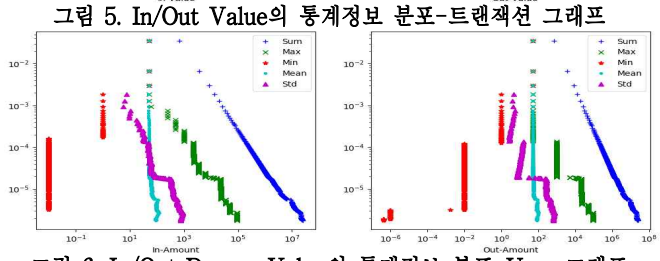


그림 6. In/Out Degree Value의 통계정보 분포-User 그래프

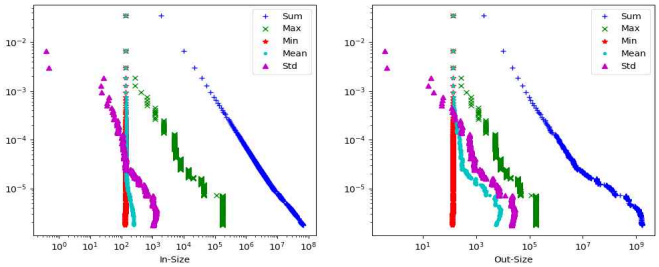


그림 7. In/Out Degree Size의 통계정보 분포-User 그래프

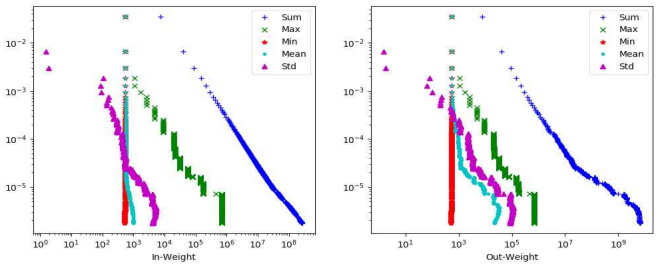


그림 8. In/Out Degree Weight의 통계정보 분포-User 그래프

그림 [5-8]은 User 및 트랜잭션 그래프의 Feature에서 통계정보를 추출하여 Degree-Stat 형태의 그래프를 나타낸 것이며 대부분의 통계정보를 이용한 그래프 분포에서 비선형적인 형태가 나타나는 것을 확인하였으며 5개의 통계정보 중 특히 표준 편차 정보에서 뚜렷한 비선형적인 형태가 나타나는 것을 볼 때 그래프 분석을 통한 이상 탐지에서 표준편차 값이 중요함을 확인하였다.

Ⅴ. 결론

본 논문은 비트코인 네트워크의 트랜잭션 데이터와 User 데이터를 수집 및 추출하고 이들의 통계정보를 노드와 에지로 이루어진 그래프 형태로 변환하고 이를 분석하였다. 분석 결과를 통해 수집할 수 있는 일반적인 정보보다 통계정보가 명확히 구분할 수 있는 분포를 띈다는 것을 확인하였다. 이를 통해 클러스터링과 같은 이상탐지를 위한 심화 분석의 가능성을 제시하였다. 향후 연구로는 K-means 알고리즘을 통해 클러스터링을 진행하고 비트코인 네트워크 내 이상 탐지에 대한 연구할 계획이며 이전 연구와의 비교를 통해 본 연구의 객관성을 갖출 예정이다.

참 고 문 헌

[1] Kalodner, Harry, et al. "BlockSci: Design and applications of a blockchain analysis platform." arXiv preprint arXiv:1709.02489 (2017).

[2] Spagnuolo, Michele, Federico Maggi, and Stefano Zanero. "Bitiodine: Extracting intelligence from the bitcoin network." International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2014.

[3] Leskovec, Jure, Jon Kleinberg, and Christos Faloutsos. "Graph evolution: Densification and shrinking diameters." ACM Transactions on Knowledge Discovery from Data (TKDD) 1.1 (2007): 2.

[4] Pham, Thai, and Steven Lee. "Anomaly Detection in the Bitcoin System-A Network Perspective." arXiv preprint arXiv:1611.03942 (2016)

Juggling Drones: Distributed Drone Port Approach to Public Drone Services

Jared Lynskey, Choong Seon Hong
{jared, cshong}@khu.ac.kr
Kyung Hee University, Suwon

Abstract

Topics related to drones are currently a hot topic among industries and academics because of their foreseen use-cases e.g disaster relief, remote search and rescue. However once drones are readily available for regular users who do not hold a pilot license, air traffic congestion and designation of safe landing areas is expected to become a challenging factor to manage. To reduce capital expenditure, we propose a system where the total number of charging stations available is less than total number of drones, hence the term juggling. A juggler must throw and catch balls in the air at the same time with two hands. Our done ports can be considered the hands with charging capabilities available to charge the drone. Our goal is to maximize the consecutive cycles of drone juggling hence minimize service interruption.

1. Introduction

Drones are becoming a hot topic in research due to their potential use cases such as first response for major disasters [1]. While, companies are offering distributed rental systems including assets like bicycles and cars, there is likely to be a system for drones in the near future too. Unlike cars or bikes drones come with their own unique challenges including, providing a safe designated area. Furthermore, in regards to the location, drone ports are expected to be small stations located on rooftops in urban areas and placed in remote areas near isolated areas such as mountains. Since there is a greater safety risk associated with drones in populated areas, expensive infrastructure is likely required. Therefore, in this paper we propose a system to allow shared drone ports in a way that is similar to juggling to reduce cost. Our proposal will provide the optimal solution to maximize the number of cycles before an interruption occurs and allow developers to disregard the initial and final location of drones completing a task.

2. Related Works

Zhang et al. (2018) considered a UAV as a tool to extend cellular coverage and alleviate communicate resource bottle necks at the edge of the network [2]. However, their system model ignores the initial and final location of the drone completing a task. This raises the question how far can the drone fly from its start location to the task and then back to a designated landing area. Kurup et al. (2015) proposed a system to sense radio-active matter in the atmosphere with drones [3]. Their objective is to minimize the distance travelled by each drone. The use of UAVs in dirty situations, such as radioactive contamination, was documented after the Fukushima reactor damage [4].

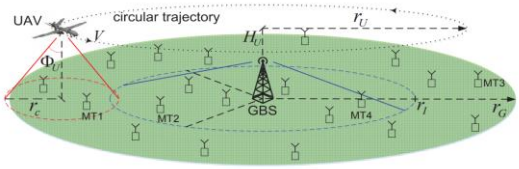


Fig. 1: UAV-aided cellular offloading.

3. System Model and Problem Formulation

Our goal is to tune the trade-off between the probability of a service interruption and drone ports capable of charging drones. For our system model we consider a set of drones denoted as $D = \{d_1, d_2, \ldots, d_n\}$, a set of Drone Ports denoted as $P = \{p_1, p_2, \ldots, p_n\}$, a set of tasks denoted as $T = \{t_1, t_2, \ldots, t_n\}$ and a set of base stations denoted as $B = \{b_1, b_2, \ldots, b_o\}$. The drone has three possible states which include, charging, flying to a task, completing a task and finally transmitting data to a nearby base station b.

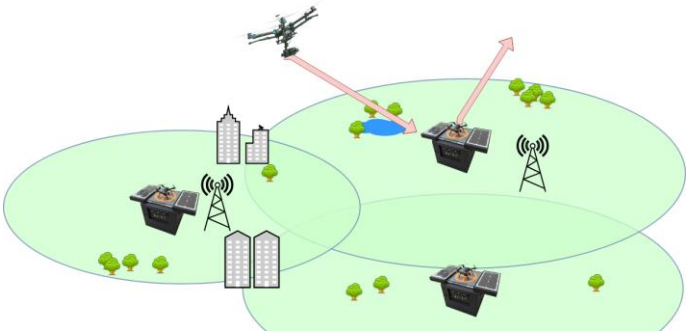


Figure 1 Juggling Act, charging drones must take off before another drone can land

Each drone d has its own energy capacity that is denoted as E = {e_1, e_2, …, e_q} Therefore, we will use e to record the current energy state of each drone. Each state excluding charging all tax the battery capacity and will be recorded by subtracting the energy consumption from the drone's. capacity. The energy by performing a task is subtracted from the drone capacity and the energy consumed by flying is denoted as gamma is subtracted

$$p_{d,s+1} = p_{d,s} - \sum_{j \in t_d} p_t - \sum_{j \in t_d} ||v^*||\gamma$$

Each task t has its own energy resource requirement to complete the task denoted as t_p. Tasks may be completed from a distance d_n creating a circular area where the drone has to enter to complete the task. This is possible since tasks such as taking images can be completed from different angles while still satisfying the request.

The access point b_o must be available to send service requests to drones. Also receive completed task data for offloading and returning data to users. We use Shannon's law to measure energy consumption and throughput.

$$r_{d,b} = W \log_2 \frac{p_d p_t}{||v_{d,t}||}$$

Drone port p is responsible for charging drones when their battery is almost depleted. Once the drone's energy is less than a threshold denoted as *L* the drone must return to the nearest drone port to be recharge. Energy is consumed while flying to and from the task, and when transmitting data to the cellular network, these may vary depending on the state of the environment. The location of these drone ports will be uniformly placed in the simulation environment and also clustered to simulate a remote and urban area. There is always equal or less drone ports than drones to reduce infrastructure cost.

Our algorithm we propose is called the drone juggler due to its behavior to juggle drones without sufficient drone ports to charge or allow drones to wait on the ground.

Algorithm 1 Drone Juggler	
Input:	Required energy from each task, Location of drone ports, location of drones, location of task, energy level of each drone
0:	**One drone is initialize in the air**
1:	**For each drone:**
2:	Assign drone to task so that energy consumed is minimized.
3:	Complete task
4:	**End for each**
5:	Central controller sends signal to drone with max power remaining to stay in the air.
6:	Drones fly to drone port with min distance and begin charging at station.
7:	**New round of tasks**
8:	Charged drones fly to complete task
9:	Drone in air lands for charging.
10:	**If drone is unable to return to base, service interruption occurs**

4. Evaluation

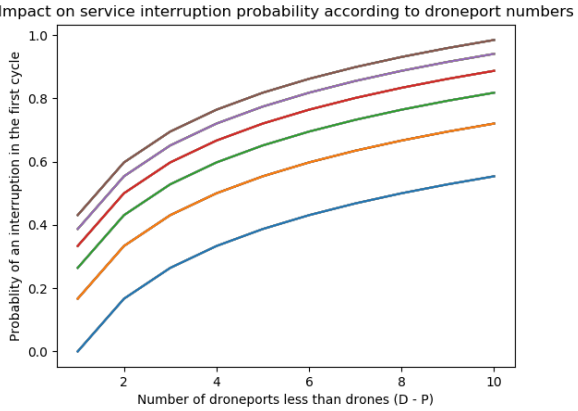
The arrival rate of tasks can be modeled as a random process. We use a Poisson process with a mean arrival rate of 1. One assumption for tasks is that they must appear in a drone's coverage area. Drones will begin at charging station with at least one drone initiated in the air due to the shortage of drone ports. Before charging, drones must be able to return back to their drone port in order to offload any data they may have gathered from the completed task. The task must be within the drone's coverage. The distance between the drone and task is calculated using co-ordinates x, y belonging to the drone and task with the Euclidean distance equation. If the distance between them is less than the threshold theta, the tasks I s assign to drone d. A charging station must be available before the drone is able to land

$$\|d\| = \sqrt{(x_d - x_t)^2 + (y_d - y_t)^2 + H^2}.$$

$$a(\|\mathbf{d}\|) = \begin{cases} 1 & for \quad 0 \geq \|d\| < \theta \\ 0 & for \quad \|d\| \geq \theta \end{cases}$$

The simulation is conducted in Python using CVXPY library. We consider a range of drones between 1 and 10. Furthermore a range of 1 to 10 tasks and finally a range of 1 to 10 drone ports. The x axis below(D–P) denotes the difference in number of drone ports and drones. Furthermore, the transmission must be fixed so that the signal can be transmitted with a minimum distance of the maximum drone threshold to remove any

We will also look at the effect of placing drone ports near base on purpose to see if there is a reduction in cost for drones. Each simulation is interrupted when a collision between two drones requesting to charge occur or when any given drone's energy is less than a given *L*. Arrival rate, flight energy and task energy are kept constant during the simulation.



Each colored line represents the relative distance between drones and tasks. From the blue line the average distance is multiplied by 1. Each line above the blue line represents the relative average distance multiplied in the order of (2x, 3x, 4x, 5x). The results show that the average distance between drone ports and tasks has a large impact on the probability that the first cycle is interrupted due to lack of charging stations. Furthermore, increasing the number of charging stations also increases the probability of a service outage.

5. Conclusion

Indeed, distributed drone ports are the future once policies are in place to support the use of autonomous drones. Our algorithm gives way to optimize the future placement of drone ports as well as number of drone ports for populated and remote areas bringing us one step closer to a true realization of autonomous drone use. Our next goal is to implement a real system applying our algorithm with open source software such as Adrudrone between the controller and drone to transmit the request requirements in real time.

6. Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2015-0-00567, Development of Access Technology Agnostic Next-Generation Networking Technology for Wired-Wireless Converged Networks) *Dr. CS Hong is the corresponding author

7. References

1. Kimon P. Valavanis and George J. Vachtsevanos. 2014. Handbook of Unmanned Aerial Vehicles. Springer Publishing Company, Incorporated.

2. J. Lyu, Y. Zeng and R. Zhang, "UAV-Aided Offloading for Cellular Hotspot," in IEEE Transactions on Wireless Communications, vol. 17, no. 6, pp. 3988-4001, June 2018. doi: 10.1109/TWC.2018.2818734

3. S. Simi, R. Kurup and S. Rao, "Distributed task allocation and coordination scheme for a multi-UAV sensor network," 2013 Tenth International Conference on Wireless and Optical Communications Networks (WOCN), Bhopal, 2013, pp. 1-5. doi: 10.1109/WOCN.2013.6616189

4. Ackerman, E. (2011) Japan Earthquake: Global Hawk UAV May Be Able to Peek inside Damaged Reactors. IEEE Spectrum.

서비스 기반 네트워크 슬라이스 선택 기능

디아즈 리베라 하비에르°, 칸 탈하 애흐마드,
메흐무드 아시프, 라픽 아딜, 송왕철
제주대학교, 컴퓨터공학과

Service based Network Slice Selection Function

Javier Diaz Rivera°, Talha Ahmed Khan, Mehmood Asif, Rafiq Adeel, Wang-Cheol SONG
Department of Computer Engineering, Jeju National University.
shaifvier@gmail.com

Abstract

Network Function Virtualization is a key enabler for the evolution of mobile networks. It drives an important feature of 5G, Network Slicing, which refers to a network aspect where multiple virtual networks can be created on top of the same physical infrastructure. Taking this into account, a mechanism that can select a slice between multiple virtual networks is required. By following the 3GPP 5G Architecture [1] a Network Slice Selection Function has been implemented in a virtual environment alongside Open Source EPC components. Working with multiple VNFs demands a medium for orchestration, due to this, the Network Slice Selection Function (NSSF) was implemented using the M-CORD Platform [2] which integrates Software Defined Networking (SDN), Network Function Virtualization (NFV), Cloud Management and a Service Orchestrator (XOS) for unifying all the network components.

I. INTRODUCTION

As Network Function Virtualization (NFV) technology matures, multiple open source initiatives have become available for network developers. Thanks to this, the days of waiting for specialized hardware to cater specific network functionality is diminishing.

NFV can be applied to multiple fields, being mobile networking one of the main areas where benefits can be achieved by the use of this technology. One of the multiple benefits that NFV brings into the table is enabling Network Slicing by allowing a physical infrastructure to be separated into multiple virtual networks that can support multiple services.

As Network Slicing is the main focus of our research, a platform capable of slicing was required. The Central Office Re-architected as a Datacenter (CORD) open source project, combines the use of Software Defined Networking (SDN), Network Function Virtualization (NFV) and Cloud Computing to successfully create multiple network slices in an E2E connection scenario. Also, by following the 5G architecture proposed by 3GPP, we have included a Network Slice Selection Function (NSSF) that is able to select between network slices by using information sent from the User Equipment (UE) as differentiator.

The CORD project has an important component called XOS, it is essential for enabling the ecosystem of multiple instances of virtual network functions (VNF). XOS defines an *extensible service control plane* that runs on top of VNF deployed in OpenStack virtual machines and SDN Applications running in ONOS.

The paper will focus on showing the implementation of the NSSF as both a VNF and an XOS Service. The scenario is running using the mobile network profile for CORD (M-CORD) as a Cord in a Box. The EPC and eNodeB are Open-Air Interface VNF developed by Eurecom [3].

This paper is organized as follows. Section II contains the system overview including the NSSF VNF and XOS Service. Section III details the implementation of the NSSF and its functionality. Section IV concludes the document.

II. SYSTEM OVERVIEW

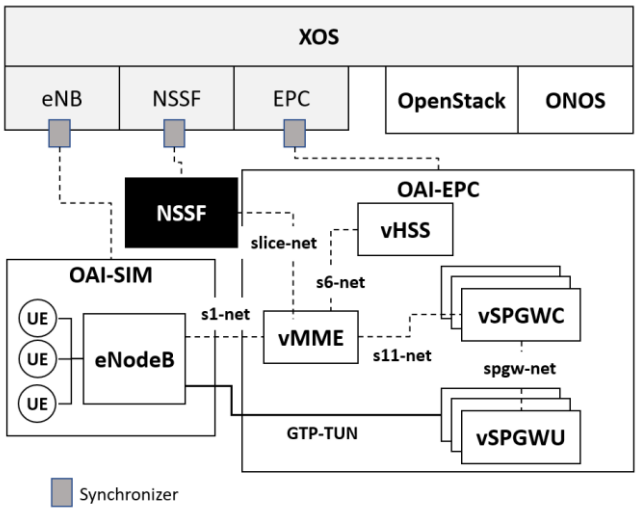


Figure 1. System Overview

Different components used in the scenario can be seen in **Figure 1**. XOS acts as the *extensible service control plane*, where multiple backend services can be created, it also acts as the Orchestrator which abstracts the interactions of the backend services with their corresponding VNF in coordination with OpenStack and ONOS. All of these components constitute the M-CORD Platform.

The eNodeB and UE are emulated by using the Open-Air Interface System Emulation (OAI-SIM) [4] and the EPC is a set of VNFs from the Open-Air Interface solution.

The backend services shown in Figure 1, have an important component that links them to their VNF, this is

called a *Synchronizer* [5]. It acts as the link between the declarative state of the system and the actual operation of it. In other words, it informs the system about the operation of the VNFs, the number of instances that have been created, if there has been an error during deployment, etc... This information is handled by each of the XOS services and through bidirectional communication, it can acquire in real time the state of the running components.

This functionality is applied to every VNF running in the E2E connection scenario.

III. NSSF IMPLEMENTATION AND FUNCTIONALITY

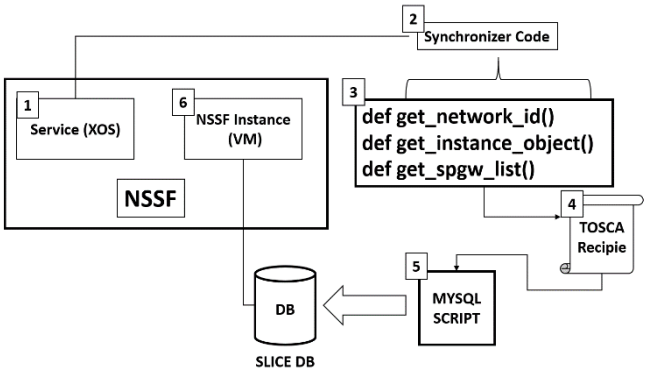


Figure 2. NSSF VNF and NSSF XOS Service

The Implementation of the NSSF is done in two parts. **Figure 2** illustrates this by showing the NSSF as both, an XOS Service and a VM Instance that acts as the VNF functionality. **1.** The role of the Service is to provide an Interface of communication between XOS, ONOS, and OpenStack. **2.** Also, and most importantly, the service has a synchronizer for the underlying system. Any changes that occur to VNF Instances are monitored by this synchronizer.

The synchronizer makes it possible to collect the current status of the mobile network (instances of each VNF that has been created in any moment). **3.** For the purpose of this research, The NSSF XOS service will monitor any SPGW that has been created, it will obtain the Id of each instance (Ip address) **4.** This information is converted into a TOSCA Recipe, which is a configuration file used in XOS. **5.** The synchronizer will take this recipe and push the information related to SPGW Instance Id into a Data Base which in turn will be accessed by the NSSF virtual machine for VNF operation. **6**.

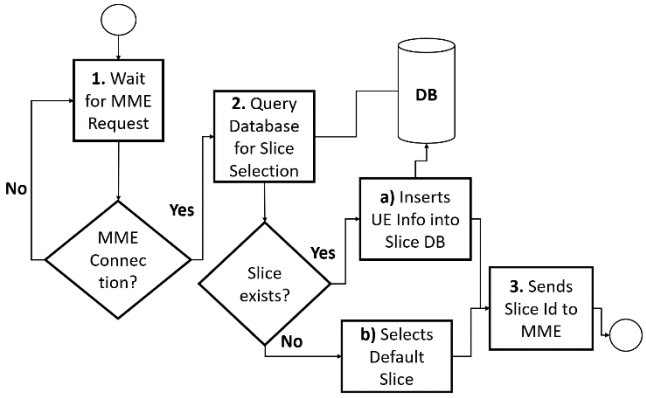


Figure 3. Slice selection procedure

Thanks to the work of the synchronizer, the process of slice selection becomes simplified. When the UE sends the connection message (PLMN, IMSI, MNC, MCC) to the vMME by means of the eNodeB emulator (OAISIM), the vMME will forwards the UE information to the vHSS for checking if it is allowed to attach and register to the network. Once the vHSS approves the UE, the vMME will forward the IMSI of the UE to the NSSF and the Slice Selection procedure will trigger.

As shown in **Figure 3**. Firstly, **1.** The NSSF VNF is always waiting for vMME connection, **2.** Once it receives the IMSI, it will verify the information in the DB and see if there is a specific vSPGW that can serve the connection.

a) If the vSPGW exists, the NSSF will store the UE Id (IMSI) alongside the slice Id that was selected
b) If it does not exist, it will select a default vSPGW for session establishment.

3. Lastly, the NSSF will reply the vMME with the Id (Ip address) of the vSPGW that will be used to serve the request.

This procedure is only possible thanks to the presence of the Orchestrator. Without XOS, the NSSF functionality would be limited, as there will be no way for knowing the status of the network in runtime.

IV. CONCLUSION

Although the NSSF fulfills a primordial functionality for a 5G network, it depends on a mechanism that can provide the state of the network and the current instantiated VNFs. By using services and synchronizers as a medium to achieve orchestration, we have bestowed the NSSF with the capability of handling slice selection in an ever-changing environment, without relying on the static information.

ACKNOWLEDGMENT

This research was one of KOREN projects supported by National Information Society Agency (16-951-00-001).

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2016R1D1A1B01016322).

REFERENCES

[1] 3GPP TS 23.501 V15.0.0 Technical Specification Group Services and System Aspects; System Architecture for the 5G System; Release 15, Dec 2017.

[2] Mobile-Central Office Rearchitected as a Datacenter (M-CORD) v4.1, [online], Available: https://guide.opencord.org/cord-4.1/ (Retrieved, October 22, 2018)

[3] Open-Air Interface Project, [online], Available: https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/home (Retrieved, October 22, 2018)

[4] Open-Air Interface System Emulation, [online], Available: https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/OpenAirLTEEmulation (Retrieved, October 22, 2018)

[5] XOS Synchronizer Framework, [online], Available: https://guide.xosproject.org/dev/synchronizers.html (Retrieved, November 6, 2018)

가상 네트워크를 위한 OpenFlow 기반 가상 게이트웨이

이도영*°, 김희곤*, 유재형† , 홍원기*

† 포항공과대학교 정보통신대학원

{dylee90, sinjint, styoo, jwkhong}@postech.ac.kr

OpenFlow-based Virtual Gateway for Virtual Networks

Doyoung Lee*, Heegon Kim*, Jae Hyoung Yoo† , James Won-Ki Hong*

*Department of Computer Science and Engineering, POSTECH

† Graduate school of Information Technology, POSTECH

요 약

네트워크 가상화는 물리 네트워크 자원을 가상화한 후 복수의 가상 네트워크를 통해 효율적으로 네트워크를 운영하도록 돕는 기술이다. 최근에는 가상 네트워크 생성 및 운영을 위해 소프트웨어 정의 네트워킹을 활용한다. 효과적인 가상 네트워크 운영을 위한 요구사항 중 하나는 가상 네트워크가 외부 네트워크와 통신할 수 있도록 외부 연결성을 가지는 것이다. 하지만 기존 SDN 기반 네트워크 하이퍼바이저들은 자체적으로 가상 네트워크의 외부 연결성을 위한 기능을 제공하지 않고, 다른 소프트웨어들과 연동을 통해 가상 네트워크를 외부 네트워크와 연결한다. 본 논문에서는 네트워크 하이퍼바이저가 외부 연결성을 지원할 수 있도록 OpenFlow 기반 가상 게이트웨이와 가상 게이트웨이 임베딩 기능을 구현하였다. 구현한 임베딩 기능은 기계학습을 활용하여 가상 게이트웨이를 위한 플로우 룰을 물리 네트워크 스위치들 사이에 분산 설치함으로써 다수의 가상 네트워크를 외부 네트워크와 연결할 때 생기는 성능 저하를 최소화한다.

Ⅰ. 서 론

네트워크 가상화 (Network Virtualization)는 효율적인 네트워크 운용을 위해 가상 네트워크를 생성하고 운용하는 기술로써 최근에는 소프트웨어 정의 네트워킹 (Software-Defined Networking, SDN)을 활용한 네트워크 가상화 기법이 주목받고 있다. SDN 기반 네트워크 하이퍼바이저 (Network Hypervisor)는 SDN 을 활용해 네트워크 가상화를 실현하는 플랫폼으로, 물리 네트워크 위에 복수의 가상 네트워크를 생성한다.

가상 네트워크를 효과적으로 운용하기 위한 요구사항 중 하나는 가상 네트워크가 외부 네트워크와 통신할 수 있도록 외부 연결성을 갖는 것이다. 하지만 기존 SDN 기반 네트워크 하이퍼바이저들은 자체적으로 가상 네트워크를 위한 외부 연결성을 지원하지 못하며, 별도의 소프트웨어 설치 및 연동을 통해 가상 네트워크와 외부 네트워크를 연결한다. 이처럼 별도로 소프트웨어를 설치하고 네트워크 하이퍼바이저와 연동하는 것은 복잡할 뿐만 아니라 네트워크 관리자에게 부담을 주는 요소이기 때문에 네트워크 하이퍼바이저가 자체적으로 가상 네트워크를 외부 네트워크와 연결할 수 있도록 기능을 제공하는 것이 필요하다.

본 논문에서는 SDN 기반 네트워크 하이퍼바이저로 생성한 가상 네트워크를 별도의 소프트웨어 도움 없이 외부 네트워크와 연결하는 OpenFlow 기반 가상 게이트웨이 (Gateway)를 제안한다. 가상 게이트웨이는 네트워크 하이퍼바이저에 의해 각 가상 네트워크의 구성 요소로 생성되며, 가상 게이트웨이를 배치한 가상 네트워크에게 외부 연결성을 제공한다. 또한, 임베딩 (Embedding) 과정에서 Feedforward Neural Network (FNN)을 활용해 가상 게이트웨이를 위한 플로우 룰 (Flow rule)을 물리 네트워크 스위치들 사이에 분산 설치함으로써 가상 네트워크의 외부 연결성으로 인한 부하를 분산시켰다.

Ⅱ. 관련 동향 및 연구

기존의 SDN 기반 네트워크 하이퍼바이저들은 생성한 가상 네트워크의 외부 연결성을 지원하기 위해 별도의 소프트웨어를 설치하고 연동한다. 예를 들어, 네트워크 하이퍼바이저로 활용 가능한 SDN 컨트롤러들인 Open Network Operating System (ONOS) [1]와 OpenDayLight (ODL) [2]은 각각 Quagga 와 OpenStack 의 Distributed Virtual Router (DVR)을 통해 가상 네트워크와 외부 네트워크를 연결한다. 하지만 Quagga 는 다수의 가상 네트워크의 외부 연결성을 지원하기 위해서는 수작업으로 복잡한 설정이 필요하며, OpenStack 은 외부 연결성만을 위해 별도로 설치하여 활용하기에는 규모나 제약 사항이 많다는 단점이 있다.

Ⅲ. 본 론

제안하는 OpenFlow 기반 가상 게이트웨이는 별도의 소프트웨어 도움 없이 네트워크 하이퍼바이저에 의해 생성된다 (그림 1). 생성된 가상 게이트웨이는 선행 연구 [3]에서 구현한 방법에 따라 ARP 를 통해 물리 게이트웨이 포트를 식별하고 가상 게이트웨이의 가상 포트를 매핑 (Mapping)한다. 가상 게이트웨이는 외부 연결성 지원을 위한 Network Address Translation (NAT) 기능과 가상 네트워크로 유입되는 트래픽을

제어하기 위한 Traffic shaping 및 Firewall 기능을 제공한다. 네트워크 하이퍼바이저는 해당 기능들을 위한 가상 플로우 룰을 생성해 가상 플로우 테이블 (Virtual Flow rule table)에 저장한다.

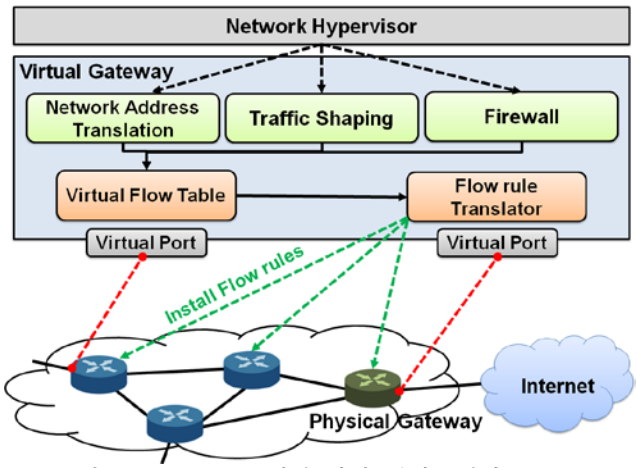


그림 1. OpenFlow 기반 가상 게이트웨이 구조

가상 플로우 룰들은 플로우 룰 변환기 (Flow rule translator)에 의해 물리 플로우 룰로 변환되어 물리 스위치들에 설치되는데, 복수의 가상 네트워크가 존재할 경우 FNN 을 통해 각 가상 네트워크 별 플로우 룰들을 분산 설치해 특정 물리 스위치에 부하가 집중되는 것을 방지한다. 이 때, FNN 은 물리 네트워크 토폴로지와 각 물리 스위치들에 저장된 플로우 룰들을 입력 값으로 받아 가상 게이트웨이 플로우 룰들이 설치될 최적의 물리 스위치를 선택한다.

OpenFlow 기반 가상 게이트웨이는 ONOS 기반 네트워크 하이퍼바이저인 ONVisor [4]에 생성 및 관리될 수 있도록 구현하였으며, 이를 위한 CLI 를 구현하였다. 또한, 가상 게이트웨이를 포함한 가상 네트워크 운영을 위해 ONVisor 에서 실행되는 Reactive forwarding application 을 추가로 구현하였다.

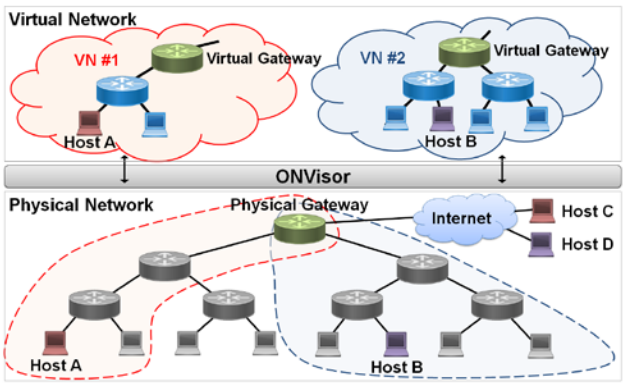


그림 2. 테스트베드 구성도

구현한 가상 게이트웨이의 기능 검증 및 성능 측정을 위해 Mininet 에뮬레이터를 활용해 테스트베드를 구축하였다 (그림 2). Mininet 으로 생성한 SDN 환경을 물리 네트워크로 가정하고, ONVisor 를 통해 가상 게이트웨이를 가지는 가상 네트워크 VN#1, VN#2 를 생성하였다. 각 가상 네트워크가 외부 네트워크와 연결이 가능함을 보이기 위해 Google 클라우드 플랫폼 서비스를 통해 외부에 가상 머신 (Virtual Machine, VM)를 생성하고, 세 가지 시나리오로 외부 연결성을 지원하여 가상 네트워크 내 VM 들과 SSH 로 연결했다 (Host A ↔ Host C, Host B ↔ Host D).

첫 번째 시나리오에서는 가상 게이트웨이 없이 물리 게이트웨이에서 수작업으로 iptables 의 설정을 변경해서 NAT 기능을 구현한 후 가상 네트워크 내 VM 들을 외부 VM 들과 연결하였다. 두 번째 시나리오에서는 FNN 을 활용하지 않은 기본 임베딩 기능으로 가상 게이트웨이를 생성했고, 세 번째 시나리오에서는 FNN 을 활용한 임베딩 기능으로 가상 게이트웨이를 생성했다. 각 시나리오에서 SSH 연결 후에는 가상 네트워크 내 VM 에서 외부 VM 에 저장된 대용량 비디오 파일을 다운로드하고, 다운로드 과정에서 Host A 와 Host B 에서 평균 Throughput 을 측정하였다.

실험 결과, 시나리오 별로 측정된 Throughput 은 각각 4.46MB/s, 2.65MB/s, 3.48MB/s 이었다. 이를 통해 가상 게이트웨이로 가상 네트워크와 외부 네트워크를 연결했을 때, Throughput 은 가상화 오버헤드로 인해 다소 떨어지는 것을 확인할 수 있었다. 하지만 이와 같은 Throughput 저하는 FNN 을 활용한 임베딩 기능으로 각 가상 게이트웨이의 플로우 룰을 분산시킴으로써 일정 부분 완화시킬 수 있는 것을 확인하였다. 결과적으로, 본 논문에서 제안하는 OpenFlow 기반 가상 게이트웨이는 비록 Throughput 은 떨어질 수 있지만 별도의 소프트웨어 설치와 수동 설정 없이 외부 연결성을 지원함으로써 네트워크 관리자의 부하를 줄일 수 있다는 장점이 있다.

Ⅳ. 결 론

본 논문에서는 가상 네트워크의 외부 연결성 지원을 위한 OpenFlow 기반 가상 게이트웨이를 제안하였다. 네트워크 하이퍼바이저는 가상 게이트웨이를 통해 자동으로 가상 네트워크를 외부 네트워크와 연결할 수 있다. 향후 연구로는 최적의 임베딩 알고리즘 연구와 가상 게이트웨이의 성능 개선이 있다.

ACKNOWLEDGMENT

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [2018-0-00749, 인공지능 기반 가상 네트워크 관리기술 개발]

참 고 문 헌

[1] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O' Connor, P. Radoslavov, W. Snow et al., " Onos: towards an open, distributed sdn os," in Proceedings of the third workshop on Hot topics in software defined networking. ACM, 2014, pp. 1- 6.

[2] J. Medved, R. Varga, A. Tkacik, and K. Gray, " Opendaylight: Towards a model-driven sdn controller architecture," in World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on a. IEEE, 2014, pp. 1- 6.

[3] Lee, Doyoung, Yoonseon Han, and James Won-Ki Hong. "Design of virtual gateway in virtual software defined networks." Network and Service Management (CNSM), 2017 13th International Conference on. IEEE, 2017.

[4] Han, Yoonseon, et al. "ONVisor: Towards a scalable and flexible SDN-based network virtualization platform on ONOS." International Journal of Network Management 28.2 (2018): e2012

SDN 기반 모바일 엣지 컴퓨팅 환경에서 머신러닝을 이용한 태스크 오프로딩 방안 연구

김기태, 홍충선*

경희대학교

glideslope@khu.ac.kr, *cshong@khu.ac.kr

A Study on Task Offloading Method Using Machine Learning in Software Defined Network Based Mobile Edge Computing

Kitae Kim, Choong Seon Hong*

*Kyung Hee University

요 약

통신기술의 발전과 다양한 모바일 기기의 개발 등으로 모바일 트래픽이 급증하는 요즘 혼잡한 네트워크를 효율적으로 관리하고 사용자들이 요구하는 서비스를 보장하기 위한 다양한 기술들이 발전되었으며 그 중 대표적인 것이 소프트웨어 정의 네트워크와 모바일 엣지 컴퓨팅이다. 본 논문에서는 SDN 기반 모바일 엣지 컴퓨팅 환경에서 제한된 자원의 모바일 기기 사용자가 엣지 노드로 태스크 오프로딩을 할 때 네트워크 환경과 엣지 노드의 자원 및 예상 실행시간을 고려한 강화학습 기반 오프로딩 방안을 제시한다.

Ⅰ. 서 론

MEC(Mobile Edge Computing)은 기존의 클라우드와 다르게 기지국이나 AP와 같은 네트워크의 가장자리에 컴퓨팅 자원을 배치시켜 사용자와 조금 더 가까운 위치에서 저 지연, 고 대역폭의 서비스를 제공하는 미래 네트워크의 핵심 기술이다. 이러한 엣지 노드는 사용자에게 미리 캐싱된 컨텐츠나 컴퓨팅 자원을 제공 할 수 있으며 이러한 서비스들은 백홀망을 거치지 않기 때문에 백홀망의 대역폭을 줄일 수 있는 동시에 사용자에게는 저 지연의 서비스를 제공할 수 있는 것이 장점이다[1-2].

최근 떠오르는 빅 데이터(Big Data), 가상현실, 증강현실 기술로 인하여 다양한 어플리케이션이 등장하였으며 이러한 어플리케이션들은 고 성능의 하드웨어 스펙을 요구하기 때문에 현재의 모바일 기기에서의 처리가 힘든 경우가 있으며 배터리 수명 또한 적절하지 않다. 하지만 통신기술의 발전으로 이러한 어플리케이션을 위한 데이터 전송이 가능케 되고 모바일 디바이스에서 모바일 엣지 노드로의 오프로딩 기술로 가능하게 되고 있다. 오프로딩 기술이란 연산에 필요한 데이터만을 엣지 노드에 전송하고 실제 연산은 엣지 노드에서 실행되고 이로부터 결과를 전송받는 기술이다. 따라서 모바일 기기는 배터리를 소모를 줄이고 빠른 어플리케이션 서비스를 이용할 수 있다. 이러한 오프로딩을 위해서는 네트워크 상황과 엣지 노드의 활용 가능한 자원들을 고려해 오프로드 시켜야 하며 해당 어플리케이션이 모바일 기기에서 실행 가능 할 시 오프로드를 시킬지에 대한 여부, 어떠한 노드로 오프로드를 시킬 것인지에 대한 활발한 연구가 진행되고 있다. 본 논문에서는 선형회귀를 통한 태스크의 수행시간, 현재 엣지 노드의 자원상황들을 고려해 강화학습 기법중 하나인 Q-Learning을 이용하여 엣지노드로 태스크를 오프로딩 하는 기법을 제안한다.

Ⅱ. 제안사항

선형회귀를 이용한 태스크 수행시간 예측

CPU_{req}	MEM_{req}	$Disk_{req}$	CPU_{cur}	MEM_{req}	$Disk_{req}$	$Time_{\exp}$
15%	44%	5%	52%	71%	12%	2.2
50%	31%	31%	41%	14%	22%	5.7
80%	23%	52%	10%	25%	48%	4.1

표 1 .태스크 수행시간 예측을 위한 데이터

표1은 각 태스크의 수행시간을 예측하기 위한 데이터 셋의 예시이다. CPU_{req},MEM_{req},$Disk_{req}$는 태스크의 자원 요구량을 나타내며 실제 수행시간과 함께 코드 프로파일링[3]을 통해 얻을 수 있다. CPU_{cur},MEM_{req},$Disk_{req}$는 실제 태스크가 실행 된 노드의 현재 자원상황이며 위와 같은 데이터로 학습된 선형회귀 모델은 실행시간 $Time_{\exp}$를 예측하기 위해 6개의 입력을 필요로 한다.

Q-Learning 기반 태스크 오프로딩

일반적으로 태스크는 오프로딩 가능한 태스크와 오프로딩 불가능한 태스크로 나뉘며 로컬 디바이스의 카메라, 센서 등을 이용하는 메소드의 경우 오프로딩 불가능한 태스크로 분류한다. 위 시스템 모델에서 모바일 유저는 SDN 컨트롤러로 서비스 요청을 보내고 SDN 컨트롤러는 각 태스크가 각 엣지 노드에서의 수행시간을 예측한다. 이러한 예측 값과 네트워크 상태를 고려해 오프로딩 할 것인지 로컬에서 수행할 것인지를 판단할 수

있다.

$$Reward = \frac{1}{Transmission\,Delay + Time_{\exp}} + Bandwidth$$

수식 2. 보상함수

수식 2는 제안하는 시스템에서 Q-Learning을 적용하기 위한 보상함수이다. 노드까지의 Transmission Delay와 예측된 수행시간이 적을수록, 그리고 대역폭이 클수록 큰 보상을 얻게 되며 반대의 경우에는 적은 보상을 받게 된다. 따라서 보상의 합을 최대로 만들기 위하여 행동하는 에이전트는 위와 같은 보상함수에 따라 오프로딩을 할 것인지 하게 되는 경우 어떤 노드로 오프로딩을 할 것인지 결정을 하게 된다.

알고리즘1. Q-Learning Based Task Offloading	
1:	**//Execution Time Prediction**
2:	en_Num = Number of Candidate Edge Node
3:	**S** -> (time[EN], Transmission Delay, Bandwidth)
4:	**A** -> (mobile device)
5:	**Task Request from User**
6:	Req($CPU_{req}, MEM_{req}, Disk_{req}, CPU_{cur}, MEM_{req}, Disk_{req}$)
7:	**for** $EN \in EN_{all}$
8:	time[EN]=prd($CPU_{req}, MEM_{req}, Disk_{req}, CPU_{cur}, MEM_{req}, Disk_{req}$)
9:	**if** time[EN] > $time_{limit}$
10:	time[EN] = -1
11:	**end if**
12:	**end for**
13:	return time[EN]
14:	**// Q-Learning Process**
15:	**while(n epochs)**
16:	Choose Edge Node Randomly with State **S**
17:	Reward = reward_function()
18:	Update Q-Function **Q(S,A)**
19:	**end while**

표 2 . 태스크 오프로딩을 위한 Q-Learning 과정

시뮬레이션

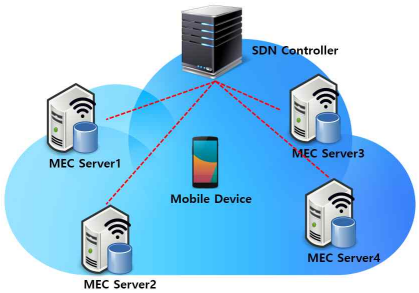


그림 1 . 시뮬레이션 토폴로지

그림2와 시뮬레이션 토폴로지는 MEC노드 4개와 모바일 디바이스 1개, SDN 컨트롤러로 이루어졌으며 MEC 노드들은 지속적으로 SDN 컨트롤러로 자신의 자원 상태 및 채널 상태를 수신한다. 본 논문에서 제안하는 사항을 검증하기 위하여 번호판 인식 프로그램[4]을 이용하였으며 3개의 임의 노드에 고의적인 혼잡상황을 주어 적절한 노드에 오프로딩이 되는지 검증하였으며 이 경우 로컬 디바이스에서의 수행시간과 클라우드 노드에서의 수행시간을 비교 측정하였다.

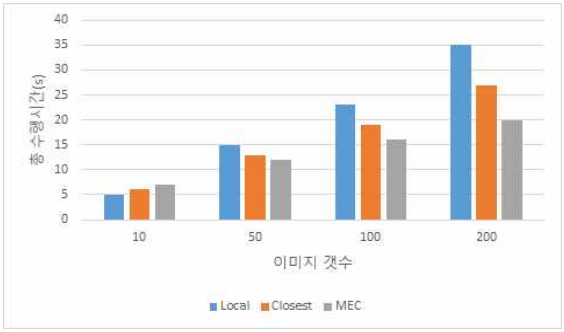


그림 2 . 실험결과

Ⅲ. 결론

본 논문에서는 SDN 기반 MEC 환경에서 모바일 디바이스의 태스크를 효율적으로 오프로드 시키기 위한 기법을 제안하였다. 성능평가 결과 네트워크 혼잡도와 리소스 사용량이 가장 적은 노드로 오프로드 되었으며 오프로드 결정이 되었을 때 로컬 디바이스에서와 가장 가까운 노드, 선택된 노드에서의 총 수행시간을 측정하였을 때 더 적게 걸리는 것을 확인 할 수 있었다. 다만 SDN 컨트롤러에서 머신러닝 및 노드의 상태 데이터를 지속적으로 받기 때문에 규모가 커지게 되는 경우 큰 부하가 발생한다. 이러한 네트워크 규모는 확실한Q-Learning의 성능 검증을 위해 필수적인 사항이므로 이러한 부하를 줄일 수 있는 연구가 필요하며 앞으로 진행 예정이다.

ACKNOWLEDGMENT

이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2015-0-00567, 유무선 통합 네트워크에서 접속 방식에 독립적인 차세대 네트워킹 기술 개발). *Dr. CS Hong is the corresponding author

참 고 문 헌

[1]윤찬현 “모바일 엣지 컴퓨팅 기술 동향”, KRnet 2018

[2] Yuyi Mao, Changsheng You, Jun Zhang, Kalbin Huang, Khaled B. Letaief, “A Survey on Mobile Edge Computing The communication Perspective”, IEEE Communications Surveys & Tutorials

[3]프로파일링,https://ko.wikipedia.org/wiki/%ED%94%84%EB%A1%9C%ED%8C%8C%EC%9D%BC%EB%A7%81_(%EC%BB%B4%ED%93%A8%ED%84%B0_%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D)

[4]OpenCV-CarLicensePlateRecognizer,https://github.com/devetude/Open-CV-CarLicensePlateRecognizier/tree/master/car_license_plate_images

SDN 기반의 OpenStack 네트워킹을 위한 Virtual TAP 설계 및 구현

정세연 [1], 유재형 [2], 홍원기 [1]

[1] 포항공과대학교 컴퓨터공학과

[2] 포항공과대학교 정보통신대학원

{jsy0906, styoo, jwkhong}@postech.ac.kr

Design and Implementation of Virtual TAP for SDN-based OpenStack Networking

Seyeon Jeong[1], Jae-Hyoung Yoo[2], James Won-Ki Hong[1]

[1]Department of Computer Science and Engineering, POSTECH

[2]Graduate School of Information Technology, POSTECH

요 약

오늘날 트래픽 규모의 증가 및 향상된 QoS(Quality of Service)에 대한 요구와 함께 클라우드 서비스가 널리 보급됨에 따라 서버 리소스의 효과적인 사용을 가능하게 하는 가상화 기술이 주목받고 있다. 본 연구에서는 기존 하드웨어 TAP(Test Access Port) 장치가 가상 링크(virtual link)를 통해 전달되는 가상 머신(Virtual Machine) 간 패킷을 복제하는데 사용될 수 없다는 문제점을 해결하기 위한 방안으로 가상 스위치에서 동작하는 Virtual TAP(vTAP)을 제안한다. 이를 위해 Port mirroring 또는 SPAN(Switched Port Analyzer)과 같은 기존 스위치의 기능을 이용할 수 있지만, 대량의 트래픽을 처리해야하는 환경(예, 데이터센터, NFV 등)에서 성능 저하 및 수동 설정에 따른 에러를 야기할 수 있다. 따라서, 본 연구에서는 ONOS(Open Network Operating System) SDN(Software-Defined Networking) 컨트롤러를 기반으로 제어되는 OpenStack 네트워크 환경에서 DPDK(Data Plane Development Kit)로 가속화된 Open vSwitch 에서 동작하는 vTAP 의 구현 및 설계를 기술하며, 제안하는 방법의 성능을 검증한다.

Ⅰ. 서 론

기존 하드웨어 TAP(Test Access Port) 장치는 EPC(Evolved Packet Core)와 같은 시스템의 각 네트워크 링크에 배치되어 통과하는 패킷을 전기적으로 복제하며, 각 TAP 장치에서 복제된 패킷은 NPB(Network Packet Broker)에서 aggregate 되어 IDS(Intrusion Detection System) 및 트래픽 analyzer 등으로 전달된다. 이러한 하드웨어 TAP 기반 패킷 모니터링 방식은 성능을 보장하지만 CAPEX 를 수반하며, 특히 오늘날 보편적인 서버 가상화 환경에서 가상 머신(Virtual Machine, VM) 및 가상 스위치 기반 가상 네트워크 환경 내부에서 발생하는 패킷을 복제하는데 사용될 수 없다. 반면, 본 연구에서 제안하는 Virtual TAP(vTAP)은 기존 TAP 장치의 소프트웨어 구현으로서, 서버 가상화 환경에서 가상 머신 간 트래픽을 패킷 수준에서 모니터링할 수 있게 한다.

우리는 기존 연구[1]에서 KVM(Kernel-based Virtual Machine) 기반의 서버 가상화 환경에서 OpenFlow 의 그룹 테이블(Group Table) 기능을 이용하여 가상 스위치에서 vTAP 을 구현하였으며, 패킷 복제 속도 측면에서 Port mirroring 기반 구현과 성능을 비교하였다. 또한, 개별 스위치 단위의 수동 설정이 필요한 Port mirroring 과는 달리, 제안된 방식은 SDN(Software-Defined Networking) 컨트롤러를 이용하여 복제 대상 패킷 플로우를 유연하게 특정짓고 중앙집중화된 방식으로 TAP 정책을 관리할 수 있음을 보였다. 본 연구에서는 기존 연구의 vTAP 설계를 확장하여 SDN 컨트롤러로 제어되는 OpenStack 네트워킹 환경(예, 데이터센터)에서 동작하도록 구현한다. 이를 통해 OpenStack VM 간 트래픽을 패킷 수준에서 모니터링할 수 있음을 보이고 실험을 통해 성능을 검증한다.

Ⅱ. 관련 연구

학계에서는 주로 데이터센터와 같은 대규모 (SDN) 네트워크의 모니터링 및 트래픽 엔지니어링을 위해 패킷 복제 기능을 활용하는 연구가 다수 존재한다 [2]. 이들 연구에서는 Port mirroring 을 통해 물리 스위치를 경유하는 패킷만 모니터링 하는 반면, 본 연구에서는 가상 스위치(OVS) 수준에서 패킷을 복제하여 호스트 서버 내부의 가상 머신 간 트래픽을 모니터링한다.

근래 네트워크 관리에 머신 러닝(Machine Learning) 기술을 접목하는 연구가 활발해짐에 따라, 일부 연구에서는 패킷 수준 데이터를 학습해 침입탐지 기능 등을 강화한다 [3]. 이를 위해 패킷 수준 데이터를 고속으로 제공하는 방안으로 본 연구의 DPDK(Data Plane Development Kit) 기반 가속화된 패킷 복제 기능이 활용될 수 있다.

Ⅲ. 제안하는 방법

그림 1 은 본 연구에서 제안하는 SDN 기반으로 운용되는 OpenStack 환경에서 VM 인스턴스 간 패킷 모니터링을 위해 OVS 가상 스위치 기반으로 vTAP 기능을 구현한 시스템의 구조를 보인다. 제안하는 구조는 크게 (1)OpenStack 네트워킹과 ONOS 컨트롤러 및 이를 연동하기 위한 ONOS 어플리케이션의 집합인 SONA(Simplified Overlay Network Architecture)와 [4], (2)vTAP 정책의 적용 및 관리를 위한 사용자 인터페이스 역할의 vTAP 어플리케이션으로 구성된다.

SONA 는 ONOS 의 주요 어플리케이션 중 하나로서, 일반적으로 Neutron 및 Open vSwitch 에이전트 기반으로

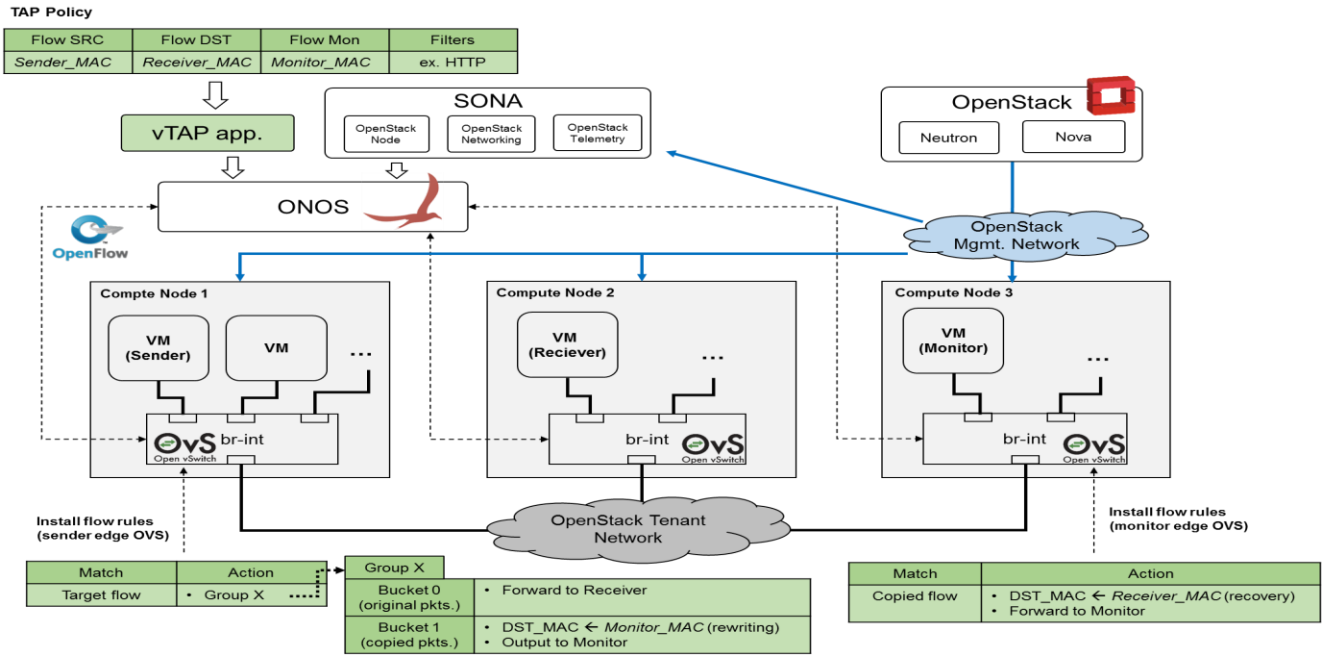


그림 1 SDN 기반 OpenStack 네트워킹에서 동작하는 vTAP 시스템 구조

동작하는 OpenStack 네트워킹을 ONOS 컨트롤러를 통해 SDN 기반으로 동작시켜 트래픽 엔지니어링 및 네트워크 정책 제어(예, vTAP)를 용이하게 한다. 본 연구에서는 SONA(ONOS 1.13 버전)와 OpenStack(Pike 버전) Nova 및 Neutron 소스 코드를 일부 수정해서 SONA 및 제안된 vTAP 어플리케이션을 통해 Compute 노드의 DPDK 기반 OVS(OVS-DPDK)에 패킷 복제 정책을 적용한다.

제안하는 vTAP 어플리케이션은 네트워크 관리자가 복제를 원하는 패킷 플로우를 출발지(source), 목적지(destination), 모니터링 목적지(monitor), 패킷 필터(filter) 수준에서 명세한 정책(그림 1 의 TAP Policy)을 ONOS API 및 OpenFlow 를 이용하여 관련 엣지(edge) 스위치(OVS)에 플로우 룰(flow rule) 형태로 반영한다. 출발지 엣지 스위치에 설치된 플로우 룰은 복제 대상 (원본) 패킷을 매칭시켜 그룹 테이블로 전달하며, 해당 그룹 테이블은 (1) 원본 패킷을 목적지로 그대로 전송하며 (2)원본 패킷을 복제하여 라우팅 정보(IP 또는 MAC 주소)를 수정, 모니터링 목적지로 전달한다. 모니터링 목적지의 엣지 스위치는 복제 패킷을 수신하여 라우팅 정보를 원상태로 복구시킨 뒤 모니터링 목적지로 전달한다.

Ⅳ. 실험

제안된 vTAP 의 성능 평가를 위해 네트워크 공격 트래픽이 유입될 때 원본 패킷 수신지(Receiver) 및 복제 패킷 수신지(Monitor)에 설치된 각 Suricata IDS 에서 생성되는 alert 의 개수를 비교하였다 (그림 2). Receiver 및 Monitor 는 2 개의 vCPU 와 2GB 메모리가 할당된 OpenStack VM 인스턴스이며, 1 개의 전용 CPU 코어가 할당된 OVS-DPDK 에서 패킷 복제 및 포워딩이 동시에 수행된다. 실험 결과에서 50 Mbps 까지는 동일한 alert 개수를 보이며 복제 패킷에 대한 IDS 분석의 정확도를 보장하지만, 100 Mbps 이상에서는 양쪽 모두 감소된 alert 개수를 보였다. 이는 분석할 패킷 개수가 증가하면서 IDS 의 리소스 점유율이 늘어나게 되고 그 결과 VM 의 패킷 drop 비율이 증가하면서 IDS 분석 정확도가 감소하였기 때문이다. 또한 패킷 복제 과정을 거치지 않고 상대적으로 빠르게 전달되는 원본 패킷을 수신하는 Receiver 에서 패킷 drop 비율이 더 높았다. 향후 연구에서 수신측 VM 에서도 DPDK 를 사용하여 패킷 처리를 위한 전용 리소스를 할당, IDS 가 사용하는 리소스와 isolation 하는 방법 등을 통해 발견된 문제점을 보완할 예정이다.

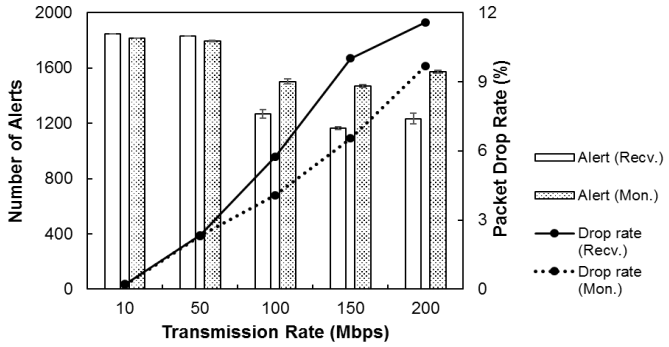


그림 2 IDS 에서 생성된 Alert 개수 비교

Ⅴ. 결론

본 논문에서는 OpenStack 환경에서 VM 인스턴스 간 패킷 모니터링을 위해 SDN 컨트롤러에서 패킷 복제 정책(추상화, 중앙집중화)을 설정하여, 가상 스위치에서 패킷 복제 기능을 수행하는 vTAP 의 구조를 설계하고 구현 과정을 설명하였다. 또한, IDS 를 활용한 유스케이스를 통해 OpenStack 환경에서의 사용 가능성과 성능을 평가하였다.

ACKNOWLEDGMENT

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [2018-0-00749, 인공지능 기반 가상 네트워크 관리기술 개발]

참 고 문 헌

[1] Jeong, Seyeon, et al. "OpenFlow-based virtual TAP using open vSwitch and DPDK." NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium. IEEE, 2018.

[2] Liu, Guyue, et al. "NetAlytics: Cloud-Scale Application Performance Monitoring with SDN and NFV." Proceedings of the 17th International Middleware Conference. ACM, 2016.

[3] Abubakar, Atiku, and Bernardi Pranggono. "Machine learning based intrusion detection system for software defined networks." Emerging Security Technologies (EST), 2017 Seventh International Conference on. IEEE, 2017.

[4] ONOS wiki, "SONA: DC Network Virtualization," 2018. [Online]. https://wiki.onosproject.org/display/ONOS/SONA%3A+DC+Network+Virtualization

xDEditor: ETSI NFV Release-3 표준 준수 Network Service Descriptor 설계 및 관리 시스템 개발

이용승, 배병관, 김동을

㈜모비젠

{ys.lee, bbangkwan, dekim}@mobigen.com

xDEditor: Development of Design and Management System for Network Service Descriptor (ETSI NFV Release-3 std. compliant)

Yongseung Lee, Byungkwan Bae, Dongeul Kim

Mobigen Co. Ltd.

요 약

네트워크 기능 가상화(NFV) 기술은 5G 이동통신의 기반 기술로써 활발한 기술개발이 이뤄지고 있다. 특히 ETSI NFV ISG를 통해 표준화가 주도적으로 행해지고 있으며, 지난 Release 2 표준을 통해 NFV MANO 내/외부에서 사용하는 모든 정보 모델과 인터페이스 규격이 정의되었다. 특히 IFA 011 및 IFA 014 표준을 통해 Network Service(NS), Virtualized Network Function(VNF)의 다양한 배포 형상을 사전에 설계하여 운용할 수 있는 데이터 모델이 정립되었다. 본 논문에서는 ETSI NFV 최신 표준(Release 3)을 준수하는 Network Service Descriptor(NSD) 설계 도구와 OSS/BSS 관점에서 개별 MANO 도메인의 NSD Lifecycle 제어를 통합 관리하기 위한 시스템을 제안한다. 제안 시스템은 복잡한 구조의 NSD를 WYSIWYG 형태의 Graphical 편집 도구를 통해 손쉬운 네트워크 서비스 설계를 가능하게 함으로써 기존 텍스트 기반 편집방식에서 문제를 해결하고 설계 시간을 효과적으로 단축한다. 또한, 통합된 NSD Lifecycle 제어방식으로 NFV 기반 차세대 이동통신망에 대한 운용효율을 극대화한다.

Ⅰ. 서 론

네트워크 기능 가상화(NFV:Network Function Virtualization)는 5G 이동통신의 기반 요소기술로써 현재 활발한 연구 개발이 진행 중인 기술 분야이다. ETSI(European Telecommunications Standards Institute)의 NFV ISG를 통해 표준화가 주도적으로 이뤄지고 있으며, 특히 IFA Working Group에서 MANO 아키텍처 내/외부 컴포넌트 간 API 및 정보 모델에 대한 표준화가 추진되었다. 지난 Release 2 발표에 포함된 IFA 014 표준에는 NFV MANO 도메인에서 Network Service를 정의하기 위한 NS Descriptor(NSD) 데이터 모델이 정의되었다[2].

NSD는 기본적으로 Network Service를 구성하는 요소인 PNF, VNF, Nested NS에 대한 정보와 기본 요소 간 네트워크를 정의하는 Virtual Link와 NS 단위의 서비스 연결점인 Service Access Point를 포함한다. 또한, 기본 요소들의 배포 형상을 Profile로 정의하고 조합을 통해 다양한 운용 환경 및 상황에 따른 배포 옵션 및 슬라이스 네트워크를 정의하는 Deployment Flavor와 VNF Forwarding Graph 정보 모델이 존재한다[2].

즉, ETSI IFA 014 표준의 NSD 데이터 모델은 기존 하나의 배포 형상 정의 방식에서 벗어나, 다수의 배포 형상을 정의할 수 있도록 고안되었다. 이를 통해, 운용자는 다양한 운용 조건 및 환경에 따라 다수의 배포 형상을 정의하고, 각 형상 간 천이 정책을 사전에 설계함으로써 실 운용 환경 변화에 자동화된 제어를 할 수 있는 기반이 마련되었다[1, 2].

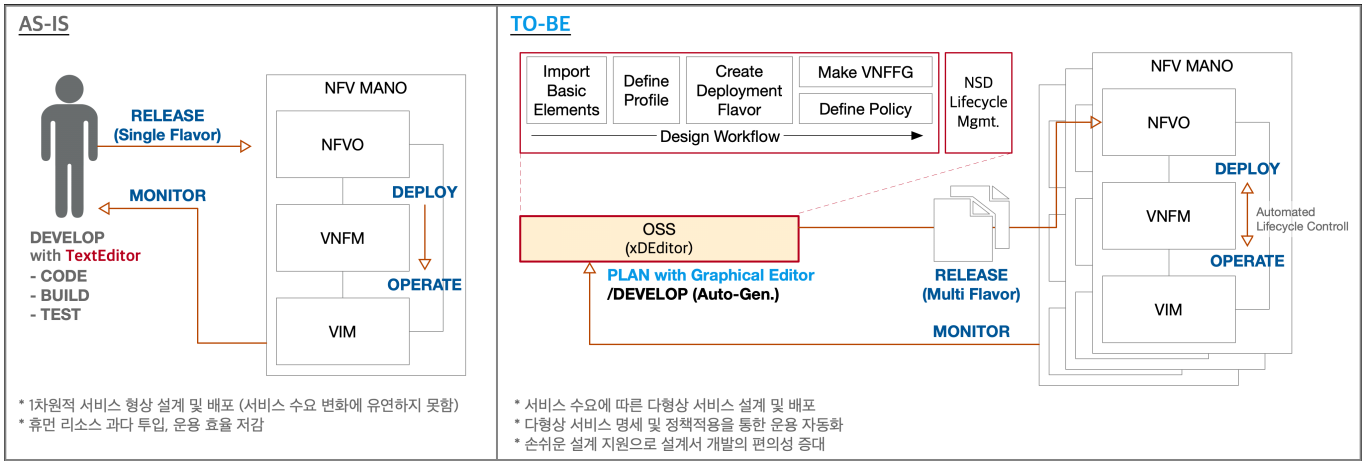
네트워크 자동화를 위해 설계단계의 역할과 업무 복잡도가 증가하였으며, Design Artifact(NSD)를 설계하고 Artifact 자체에 대한 Lifecycle 관리를 수행하는 Design-Time 역할과 Artifact를 이용하여 실제 서비스를 운용하는 Run-Time 역할로 기능이 분리되었다. 하지만 대부분의 Opensource Project(Opensource MANO:OSM, Open Baton, ONAP 등)에서는 실제 NFV 운용을 실현하기 위한 Run-Time 기능 연구 개발에 주로 초점이 맞춰져 있으며 Design-Time 기능에 대한 솔루션은 큰 성과가 없는 상태이다[표1]. 특히 NFVO 에 해당하는 Open Baton 및 OSM의 경우 Flavor 정보 모델에 대한 기능이 없거나 미약하며 NSD를 Text Editing을 통해 작성해야 하는 불편함이 있다.

	xDEditor	OPEN BATON	OSM	ONAP
ETSI 표준 - 기본모델	Rel.3	Rel.2	Rel.2	Own Spec.
ETSI 표준 - Flavoring	Rel.3	X	Rel.1	Own Spec.
Network Slice 설계	O	X	X	X
GUI Editor	O	X	X	O
Self-Service	O	X	X	TBD
정책 설정	O	X	X	TBD
배포관리	O	X	X	TBD
Run-time 기능	X (타 Subsystem 담당)	O	O	O

〔표 1〕 Opensource Project의 Design Time 기능 구현 현황 및 제안 시스템의 목표 기능

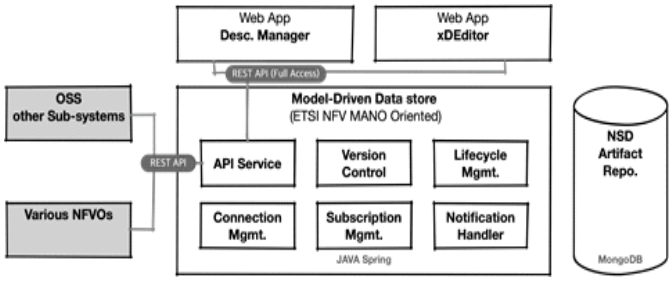
본 연구를 통해 ETSI NFV MANO 최신 표준 규격을 준수하는 Network Service Descriptor (NSD) 설계 도구를 제안한다. 설계 도구는 그래픽 편집기 형식으로 복잡한 NSD 설계를 직관적이고 쉽게 수행하도록 지원한다. 또한, NSD에 대한 버전관리를 자체적으로 수행하고 NFVO와 연동을 통해 배포 및 Lifecycle 관리를 통합적으로 수행한다. 제안 시스템의 구현 범위 및 목적은 다음과 같다.

- 표준 NSD Information Model(IFA014) 기반의 설계 도구 (Graphical)
- Os-Ma-Nfvo reference point (IFA013) 구간의 NSD, VNFD Lifecycle(On-board/Update/Delete) 관리 도구 (API Service)



〔그림 3〕 제안 시스템을 통한 Network Service Planning 지원으로 NFV MANO 환경의 DevOps 실현

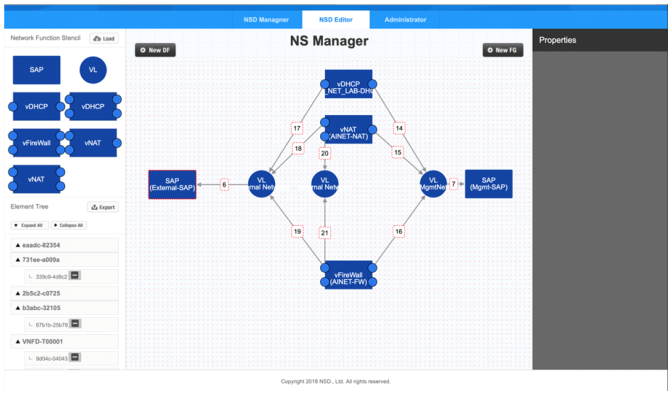
Ⅱ. 본론



〔그림 1〕 제안 시스템의 Software 컴포넌트 구조

제안 시스템은 [그림 1]과 같이 구성된다. 전통적인 WEB-WAS-DB 3 tier 구조를 갖는다. NSD에 대한 저작 및 Lifecycle 관리를 위한 REST API 서비스를 제공한다[3, 4]. WEB Application으로는 GUI를 이용한 NSD 모델에 대한 편집 기능을 제공하는 xDEditor [그림 2]와 NSD Lifecycle 통합 관리를 수행하는 Descriptor Manager로 구성된다. xDEditor를 이용한 NSD 설계 workflow는 다음과 같다 [그림 3].

1. **ImportBasicElements**: NS에 포함할 기본 요소(VNF, PNF, Nested NS)를 불러와 Stencil을 구성한다. 공통 요소인 SAP, Virtual Link는 기본으로 포함되어있다. 이후 Stencil에 등록된 Element를 Canvas 영역으로 Drag&Drop 하여 위치하고, Element 간 연결 관계를 정의한다.
2. **Define Profile**: 각 Element의 Flavor와 Instantiation Level을 정의하고 Scaling을 위한 Instance 수를 정의한다.
3. **Create Deployment Flavor**: Flavor에는 다 수의 Instantiation Level을 정의할 수 있으며, 각 Level에 해당하는 Elements의 Profile 매핑 통해 배포 형상을 정의한다.
4. **Define State Policy**: Monitoring Parameter에 정의된 Metric을 이용하여 조건을 정의하고, 각 조건에 맞는 Flavor 및 Instantiation Level을 지정하여 상황별 천이 정책을 정의한다.
5. **Make VNFFG**: 각 Flavor 별 토폴로지 화면에서 개별 슬라이스에 포함될 요소를 선택하여 그룹화하여 VNFFG를 정의한다.



〔그림 2〕 xDEditor의 NSD 편집 화면

6. **NSDLCM**: 설계가 완료된 NSD를 연동 중인 NFVO에 On-boarding, Enabling, Disabling Deleting 등 Lifecycle Operation을 수행한다.

연동 중인 NFVO에 의해 NSD의 Lifecycle 상태가 변경된 경우, Notification을 받아 상태정보를 동기화 할 수 있다. 또한, xDEditor를 통해 설계된 NSD는 WAS-DB에 저장되어 Version Control 기능을 제공하여 설계 물에 대한 변경 추적이나 버전 관리를 자체적으로 수행한다.

Ⅲ. 결론

본 논문의 제안 시스템은 ETSI NFV의 최신 표준을 준수하는 NSD의 저작 기능과 Lifecycle 관리기능 수행한다. 기존의 텍스트 기반 NSD 저작 방식보다 Graphical 한 도구를 이용해 복잡한 정보 모델을 효과적으로 설계할 수 있도록 지원한다. 이를 통해 운용자의 설계 시간 단축과 휴먼 에러를 미연에 방지할 수 있다. 또한, Design Artifact인 NSD에 대한 Lifecycle 관리를 통합하여 지원하므로, 다수의 NFVO를 제어하는 OSS 관점에서 개별 NFVO 별 배포 상태나, 버전 관리의 편의성을 제공한다. 본 시스템을 통해 다형상 배포에 대한 정의와 운용 정책에 대한 Planing을 지원하여 NFV 운용 환경에서의 DevOps 및 운용 자동화의 진정한 실현을 기대할 수 있다[그림 3].

본 시스템은 NFV의 Design-Time 기능을 독립적으로 수행한다. 따라서 NFVO 혹은 OSS의 Sub-system으로 Integration 되어 동작할 수 있다. 향후 Open Source Project의 업스트림 일정에 따라, ETSI release-3 표준을 지원하는 NFVO에 우선하여 연동 및 통합 개발을 수행할 계획이다.

ACKNOWLEDGMENT

본 연구 개발은 한국정보화진흥원의 2018년 NET 챌린지 캠프 시즌5 챔피언스리그 사업[2018-0-00593-001]의 지원으로 수행되었음

참 고 문 헌

[1] ETSI NFV ISG, “GS NFV-IFA 013 v3.1.2 NFV Management and Orchestration Os-Ma-Nfvo reference point Interface and Information Model Specification,” 2018.

[2] ETSI NFV ISG, “GS NFV-IFA 014 v3.1.2 NFV Management and Orchestration Network Service Templates Specification,” 2018.

[3] ETSI NFV ISG, “GS NFV-SOL 001 v0.12.0 NFV Release 2 Protocols and Data Models NFV descriptors based on TOSCA specification,” 2018.

[4] ETSI NFV ISG, “GS NFV-SOL 005 v2.5.2 NFV Release 2 Protocols and Data Models RESTful protocols specification for the Os-Ma-nfvo Reference Point,” 2018.