http://www.knom.or.kr

ISSN : 2586-0232

2019년 통신망운용관리 학술대회 논문집

Proceeding of KNOM Conference 2019

- 일시 : 2019년 5월 30일(목) ~ 31일(금)
- 장소 : 계명대학교 행소박물관, 대구광역시
- 주관 : 한국통신학회, 통신망운용관리연구회
- 주최 : 계명대학교

한국통신학회 통신망운용관리연구회

초대의 말씀

한국통신학회 통신망운용관리 연구회 (KNOM)는 2019년 통신망 운용관리 학술대회 (KNOM Conference 2019)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 튜토리얼 및 연구 논문을 모집합니다.

네트워크와 컴퓨팅 기술은 최근 급속하게 발전하고 변화하고 있습니다. 날로 고속화되는 무선기술과 차세대 인터넷에 대한 지대한 관심은 이러한 네트워크 기술 발전을 대변하고 있습니다. 또한 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 연구주제였던 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 보안침해도 급격히 증가하여 통신망 운용관리 분야의 연구와 개발 범위 또한 급속히 넓어지고 그 중요성이 더욱 강조되고 있습니다.

이러한 추세를 반영하여 KNOM Conference 2019에서는 통신망 전반에 대한 모델링, 설계, 서비스 제공, 운용 관리 및 보안 기술 분야의 최신 연구 개발 결과에 대한 유익한 정보제공과 토론의 장을 제공할 계획입니다. 부디 본 Conference가 운용관리 전반에 걸친 활발한 기술교류와 토론의 장이 될 수 있도록 각 분야의 전문가 여러분들의 적극적인 논문투고와 발표를 기대합니다.

또한 본 학술대회에 통신망 운용관리와 관련된 연구소, 학계, 통신서비스 사업자, 통신기기 제조업체 등에서 많은 분들이 참석하여 실제적인 기술을 습득하고 토론할 수 있는 좋은 기회가 될 수 있기를 바랍니다.

2019. 05. 30

2019 통신망운용관리학술대회 운영위원장 석승준
한국통신학회 통신망운용관리연구회 위원장 주홍택

운영위원회

운영위원장	석승준 (경남대)	**학술**	원영준 (한양대)
포스터	조부승 (KISTI)	**튜토리얼**	김명섭 (고려대)
초청강연	석우진 (KISTI)	**홍보 및 출판**	김경백 (전남대)
등록 및 예산	최미정 (강원대)	**현장**	주홍택 (계명대)
전시	최태상 (ETRI)		
자문	홍충선(경희대), 홍원기(POSTECH), 최덕재(전남대), 김영탁(영남대), 이영우(MOS강서), 유재형(POSTECH), 송왕철(제주대)		

프로그램

시 간	주요 행사
2019년 5월 30일 (목)	
발표장소	행소박물관 시청각실
12:00 ~ 13:00	등록
13:00 ~ 14:20	**TS1. 블록체인 및 IoT 시스템 관리** (좌장: 주홍택 교수, 계명대)
14:20 ~ 14:50	Poster Session 1. (좌장: 조부승 박사, KISTI)
14:50 ~ 15:00	Coffee Break
15:00 ~ 15:20	개회식 I 사회: 김명섭 교수(고려대) 개회사 I KNOM 2019 학술대회위원장 석승준 교수(경남대) 축 사 I 통신망운용관리연구회위원장 주홍택 교수(계명대)
15:20 ~ 16:00	**[Keynote] 블록체인 발전방향(김종현PM [IITP])** (사회: 주홍택 교수, 계명대)
[Tutorial Session]	
16:20 ~ 17:20	Cloud Native Networking 개요 및 SONA-CNI 개발현황 (리건 박사, SKT)
19:00 ~	**만찬**

프로그램

시 간	주요 행사
2019년 5월 31일 (금)	
발표장소	행소박물관 시청각실
09:00 ~ 09:30	등록
09:30 ~ 10:50	**TS2. 분산 및 에지 컴퓨팅 관리** (좌장: 김경백 교수, 전남대)
10:50 ~ 11:00	Coffee Break
11:00 ~ 12:20	**TS3. 미래 네트워크 관리** (좌장: 최미정 교수, 강원대)
12:20 ~ 13:30	Lunch
13:30 ~ 14:50	**TS4. 네트워크 보안 관리** (좌장: 석승준 교수, 경남대)
14:50 ~ 15:20	Poster Session 2. (좌장: 김윤희 교수, 숙명여대)
15:20 ~ 15:30	Coffee Break
[Invited Speaker Session] 좌장:석우진 센터장(KISTI)	
15:30 ~ 16:10	**초청세미나1:** 양자컴퓨팅의 과거, 현재 그리고 미래 최명수 박사(ETRI)
16:10 ~ 16:50	**초청세미나2:** 양자암호통신 기술이슈와 동향 장진각 박사(국가보안기술연구소)
16:50 ~ 17:30	**초청세미나3:** 딥러닝기반 대도시 교통흐름 예측 및 신호시스템 개발 이홍석 박사(KISTI)
17:30 ~	[Best Paper Award & 경품 추첨*] 폐회사
*경품추첨안내: 본인 미 참석일 경우, 경품을 수령하실 수 없습니다.	
18:00 ~ 20:00	KNOM OC Wrap-up Meeting

논문 목차

Technical Session 1 [5월30일(목) 13:00 ~ 14:20]

1. 비트코인 트랜잭션 수 예측을 위한 **LSTM** 학습데이터 선택기법, *지세현, 구영훈, 백의준, 박지태, 윤성호, 김명섭 (고려대학교, LG전자)* ------ 1 ~ 3

2. 비트코인 데이터의 분석 및 시각화를 위한 웹 서버 구축, *신혜영, 김대용, 이기용, 주홍택 (계명대학교)* ---------- 4 ~ 7

3. **IoT** 플랫폼을 위한 **MQTT** 클러스터에서 서브스크라이버 배정 방안, *강귀영, 석승준 (경남대학교)* ---------- 8 ~ 11

4. **End-to-end Network Resource Slicing in Mobile Edge Computing for 5G New Radio**, *Yan Kyaw Tun, Shashi Raj Pandey, 홍충선 (경희대학교)* ---------- 12 ~ 15

5. 비트코인 노드 메모리 풀 유사도 분석, *고경찬, 이채현, 홍원기 (포항공과대학교)* ---------- 16 ~ 18

Technical Session 2 [5월31일(금) 09:30 ~ 10:50]

6. 강화학습을 이용한 **UAV-EDGE** 협업 태스크 오프로딩 방안 연구, *김기태, 홍충선 (경희대학교)* ---------- 19 ~ 22

7. **Caching in Named Data Networking with an Open Source Edge Computing Framework**, *Muhammad Atif Ur Rehman, Rehmat Ullah, 김병서 (홍익대학교)* ---------- 23 ~ 24

8. 우주 기상 관측 데이터를 위한 분산 저장소 요가안드리안, 주홍택 (계명대학교) ---------- 25 ~ 30

9. 컨테이너 기반 **M-CORD** 모니터링 시스템 설계 및 구현 홍지범, 김우중, 유재형, 홍원기 (포항공과대학교) ---------- 31 ~ 32

10. **CUDA** 다중 프로세스 실행 서비스를 이용한 과학 응용 실행 패턴 분석 김세진, 오지선, 김윤희 (숙명여자대학교) ---------- 33 ~ 36

논문 목차

Technical Session 3 [5월31일(금) 11:00 ~ 12:20]

11. VNF를 위한 자동화된 스케일링 응용과 마이크로서비스 기반 모니터링, *아시프 매흐무드, 송왕철 (제주대학교)* ---------- 37 ~ 39

12. 인공지능 기반 NFV 관리 시스템 구조 및 테스트베드 구축, *정세연, 이도영, 유재형, 홍원기 (포항공과대학교)* ---------- 40 ~ 42

13. 5G MEC 환경에서의 종단간 지연 시간 측정 기법 연구, *현종환, 유재형, 홍원기 (포항공과대학교)* ---------- 43 ~ 45

14. SDN 기반 QoS 보장형 Fast BSS Transition, *황현동, 김영탁 (영남대학교)* ---------- 46 ~ 49

15. IEEE 802.11ah/ Sub-1GHz 기반의 사물인터넷 스마트 제어, *김민철, 김영탁 (영남대학교)* ---------- 50 ~ 53

Technical Session 4 [5월31일(금) 13:30 ~ 14:50]

16. 네트워크 보안을 위한 SIEM 솔루션 비교 분석, *이종화, 방지원, 김종욱, 최미정 (강원대학교)* ---------- 54 ~ 57

17. Manual Unpacking을 위한 Anti-Debugging 무력화에 관한 연구, *김종욱, 방지원, 최미정 (강원대학교)* ---------- 58 ~ 61

18. 유해 네트워크 트래픽 탐지를 위한 컨볼루션 신경망기반 트래픽 분류 기법 연구, *염성웅, 뉘엔 지앙 쓰엉, 뉘엔 반 퀴엣, 김경백 (전남대학교)* ---------- 62 ~ 64

19. 활성 사용자 지표 기반 에듀롬 인증로그 분석, *장민석, 조부승 (한국과학기술정보연구원)* ---------- 65 ~ 68

20. 확장성있는 이벤트 알림 봇 설계 및 구현, *문현수, 이영석 (충남대학교)* ---------- 69 ~ 72

Poster Session 1 [5월30일(목) 14:20 ~ 14:50]

21. M-CORD 모니터링 시스템을 이용한 비정상 상태 탐지 연구, *박수현, 홍지범, 유재형, 홍원기 (포항공과대학교)* ---------- 73 ~ 74

22. 네트워크 텔레메트리를 활용한 머신 러닝 기반 네트워크 이상 탐지 기법 연구, *남석현, 현종환, 유재형, 홍원기 (포항공과대학교)* ------- 75 ~ 77

23. 인공지능 기반 VNF 자원 예측 모델 연구, *김희곤, 정세연, 유재형, 홍원기 (포항공과대학교)* ---------- 78 ~ 79

24. 가청 주파수를 이용한 Hello 보내기, *김수현, 문현수, 이영석 (충남대학교)* ---------- 80 ~ 82

25. 머신러닝 기반 동적 경로 가중치 조정 로드밸런싱 알고리즘 연구, *임지윤, 현종환, 유재형, 홍원기 (포항공과대학교)* ---------- 83 ~ 86

26. Modbus/TCP 프로토콜 기반 클러스터링 알고리즘 성능 평가, *이민성, 심규석, 이민섭, 박준상, 김명섭 (고려대학교)* ---------- 87 ~ 88

27. 차세대 국제 연구망 구축을 위한 네트워크 인프라 설계 요건에 관한 연구, *조부승, 장민석 (한국과학기술정보연구원)* ---------- 89 ~ 91

28. 고신뢰성 5G 코어망을 위한 네트워크 자동화 기법 연구, *이재욱, 고한얼, 이호찬, 백상헌 (고려대학교)* ---------- 92 ~ 93

Poster Session 2 [5월31일(금) 14:50 ~ 15:20]

29. De-anonymizing Bitcoin through mapping Transactions and public-keys to nodes Meryam Essaid, *박세진, 주홍택 (계명대학교)* ---- 94 ~ 98

30. 비트코인 네트워크의 불법거래 탐지 연구, *이채현, Sajan Maharjan, 고경찬, 홍원기 (포항공과대학교)* ---------- 99 ~ 101

31. 주성분 분석을 적용한 클러스터링을 이용한 비트코인 네트워크 분석 방법, *신무곤, 백의준, 구영훈, 지세현, 박준상, 김명섭 (고려대학교, LG전자)* ---------- 102 ~ 103

32. Named Data Networking with Edge Computing for 5G Radio Access Network, *Rehmat Ullah, Muhammad Atif Ur Rehman, 김병서 (홍익대학교)* ----------104 ~ 105

33. 혼잡한 대중교통 상황에서의 p2p offloading 기법 연구, *백호성, 백상헌 (고려대학교)* ----------106 ~ 107

34. 허가형 블록체인을 이용한 마이크로 그리드 에너지 공유 프레임워크, *전정민, 강선무, 홍충선 (경희대학교)* ----------108 ~ 110

35. 합의 알고리즘 성능 검증을 위한 블록체인 네트워크 시뮬레이터 모델, *강창훈, 고경찬, 홍원기 (포항공과대학교)* ---------- 111 ~ 113

36. 머신 러닝을 이용한 비트코인 트랜잭션 수수료 예측, *최원석, 고경찬, 홍원기 (포항공과대학교)* ----------114 ~ 116

비트코인 트랜잭션 수 예측을 위한 LSTM 학습데이터 선택기법

지세현, 구영훈, 백의준, 박지태, 윤성호*, 김명섭
고려대학교, *LG전자
{sxzer, gyh0808, pb1069, pjj5846, tmskim}@korea.ac.kr, *sungho.sky.yoon@lge.com

LSTM Learning Data Selection Technique for Number of Bitcoin Transactions Prediction

Se-Hyun Ji, Young-Hoon Goo, Ui-Jun Baek, Jee-Tae Park, Sung-Ho Yoon*,
Myung-Sup Kim
Korea Univ. *LG Electronics

요 약

블록체인 기술을 기반으로 만들어진 온라인 암호화폐인 비트코인은 오늘날 개인, 기업, 정부, 금융기관 등 모두의 관심을 끌고 있다. 지난 몇 년간 비트코인 트랜잭션 수가 증가함에 따라 비트코인 시장의 규모는 나날이 증가하고 있다. 비트코인 트랜잭션 수를 예측하는 것은 비트코인 네트워크에 있어 중요한 항목이다. 본 논문은 비트코인 트랜잭션 수를 예측하기 위해 기계학습 알고리즘 중 하나인 LSTM 모델의 학습데이터 선택기법을 제안한다. 본 연구팀이 수집한 비트코인 블록의 트랜잭션 통계데이터와 해당 블록에 담긴 트랜잭션 수의 상관분석 방법을 적용하여, LSTM 모델의 학습데이터로 선정한 뒤, LSTM 모델이 예측한 값과 실제 값의 차이를 비교하여 제안하는 기법의 타당성을 검증한다.

Ⅰ. 서론

사토시 나카모토에 의해 개발된 비트코인은 블록체인 기술을 기반으로 만들어진 온라인 암호화폐이다[1]. 현재 비트코인은 정부, 기업, 금융기관 등 모두의 관심을 끌고 있다. 지난 몇 년간 비트코인의 트랜잭션 수는 엄청나게 증가하고 있다. 2019년 4월 기준, 비트코인의 시가 총액은 약 100조 원에 달한다. 비트코인 트랜잭션 수의 증가에 따라 비트코인 네트워크는 급속도로 발전을 하고 있지만, 그에 따른 문제도 발생하고 있다. 예를 들어 트랜잭션 처리 비용은 증가했지만, 트랜잭션 확인 시간은 지연되고 있다. 이러한 이유로 미래의 비트코인 트랜잭션 수를 예측하는 것은 비트코인 네트워크의 성장 및 문제에 대응하는 것에 있어 중요하다[2]. 그러나 현재 비트코인 트랜잭션 수를 예측하기 위한 연구는 거의 없다.

기계학습 알고리즘 중 하나인 Long Short Term Memory (LSTM)은 순환신경망 구조로부터 파생된 시계열 데이터를 예측하는 데 있어 특화된 알고리즘이다[3]. 일정 시간 간격으로 생성되는 비트코인 블록데이터는 시계열 데이터로써 LSTM 모델의 학습데이터로 적합하다. 그러나 비트코인 트랜잭션의 수를 예측하는데 어떤 항목의 비트코인 블록데이터를 학습해야 하는지는 알 수 없을 뿐만 아니라 적합한 학습데이터를 찾기 위해 모든 경우의 수를 이용하여 데이터를 학습하는 것은 비효율적이다. 따라서 LSTM 알고리즘을 적용한 비트코인 트랜잭션 수를 예측하는데 적합한 학습데이터를 찾는 효율적인 방법이 필요하다.

본 논문은 서론에서 연구 배경과 목표를 서술하고, 본론에서 비트코인 트랜잭션 수 예측을 위해 기계학습 알고리즘 중 하나인 LSTM 모델의 학습데이터를 선정하기 위한 2가지의 상관관계 분석을 제안한다. 제안하는 방법은 실험을 통해 타당성을 검증한다.

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

Ⅱ. 본론

본 장에서는 비트코인 블록 통계데이터와 비트코인 트랜잭션 수와의 상관분석 방법 및 본 연구팀이 수집한 비트코인 블록 통계데이터와 상관분석 방법을 적용하여 선정된 실험데이터에 대해 언급한다.

2.1 상관분석

본 절에서는 상관분석 방법에 대해 언급한다. 상관분석은 두 변수 사이에 어떤 선형적 관계가 있는지를 분석하는 방법이다. 두 변수는 서로 독립적인 관계이거나 상관된 관계일 수 있으며 이때 두 변수 사이의 관계 강도를 상관관계라 한다.

표 1. 피어슨 상관계수 해석표

상관계수 r의 범위	해석
$-1.0 \le r \le -0.7$	강한 음의 선형적 관계
$-0.7 \le r \le -0.3$	뚜렷한 음의 선형적 관계
$-0.3 \le r \le -0.1$	약한 음의 선형적 관계
$-0.1 \le r \le +0.1$	무시 될 수 있는 선형적 관계
$+0.1 \le r \le +0.3$	약한 양의 선형적 관계
$+0.3 \le r \le +0.7$	뚜렷한 양의 선형적 관계
$+0.7 \le r \le +1.0$	강한 양의 선형적 관계

본 논문에서는 2가지의 상관분석 방법을 이용하였다. 첫 번째 상관분석 방법은 피어슨 상관계수(Pearson correlation coefficient)를 이용하는 것이다. 피어슨 상관계수는 두 변수 사이의 선형 상관관계를 계량화한 수치다. 피어슨 상관계수는 코시-슈바르츠 부등식에 의해 +1과 -1 사이의 값

을 가지며, +1은 완벽한 양의 선형적 상관관계, 0은 선형 상관관계가 없음, -1은 완벽한 음의 선형적 관계를 의미한다[4]. 피어슨 상관계수에 대한 해석은 표1과 같다. 두 번째 상관분석 방법은 스피어만 상관계수(Spearman correlation coefficient)를 이용하는 것이다. 스피어만 상관계수는 자료의 값 대신 순위를 이용하는 경우의 상관계수로써, 데이터를 작은 것부터 차례로 순위를 매겨 서열 순서로 바꾼 뒤 순위를 이용해 상관계수를 구한다. 스피어만 상관계수는 두 변수 사이의 연관 관계가 있는지 없는지를 밝혀준다. 스피어만 상관계수의 값이 -1과 +1 사이의 값을 가지는데 두 변수 안의 순위가 완전히 일치하면 +1이고, 두 변수의 순위가 완전히 반대이면 -1이 된다[5].

2.2 비트코인 블록 통계데이터

본 절에서는 비트코인 블록의 트랜잭션 통계데이터에 대해 언급한다. 비트코인 네트워크의 100,000 높이의 블록부터 200,000 높이의 블록에 담긴 트랜잭션 데이터를 수집하였고, 수집한 트랜잭션 데이터를 재정렬하여 80가지 항목의 트랜잭션 통계데이터를 추출하였다. 추출한 비트코인 트랜잭션 통계데이터 항목은 표 2와 같다. 비트코인 데이터를 블록, 트랜잭션 단위로 구분하였고 각 데이터 항목에 대한 합, 평균, 최댓값, 최솟값, 표준편차이다.

표 2. 비트코인 트랜잭션 통계데이터 항목

비트코인 트랜잭션 통계데이터 항목		
단위	**항목**	**통계정보**
Block	Tx Size	Sum
	Tx vSize	Mean
	Fee	Max
	Value	Min
	n_Vin	Stdvar
	n_Vout	
Transaction	Input	
	Output	

2.3 실험데이터

본 절에서는 비트코인 블록 통계데이터를 2가지 상관분석을 적용하여 선정된 실험데이터에 대해 언급한다. 80가지 항목의 비트코인 통계데이터와 비트코인 트랜잭션 수와의 상관분석을 하였다. 80가지의 비트코인 통계데이터 항목 중 상관분석을 적용하여 비트코인 트랜잭션 선정된 실험데이터 항목은 그림1, 2와 같이 상관계수 값의 크기에 대해 오름차순으로 나열하였다.

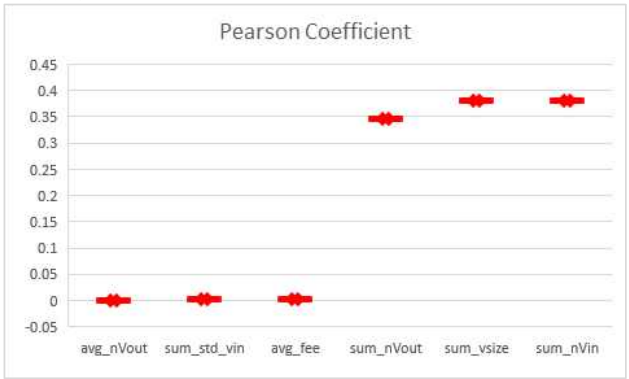


그림 2. 피어슨 상관계수 실험데이터 항목

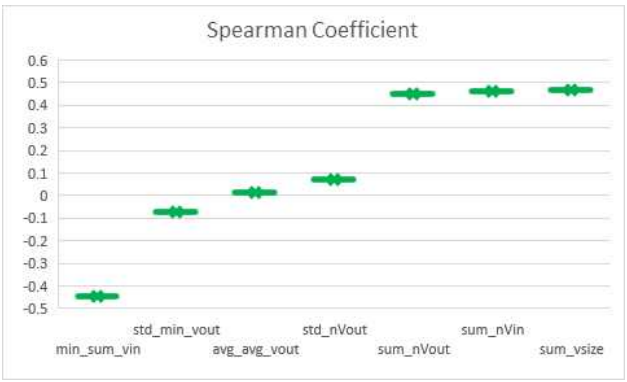


그림 1. 스피어만 상관계수 실험데이터 항목

피어슨 상관계수 분석을 적용한 경우 블록 당 비트코인 트랜잭션의 수와 양의 선형적 관계가 있는 데이터 3개와 성능 비교를 위해 선형적 관계가 없는 데이터 3개를 선정하였다. 음의 선형적 관계를 나타내는 데이터는 없었다. 스피어만 상관계수 분석을 적용한 경우 -1 또는 1에 가까운 4개의 데이터와 성능 비교를 위해 상관관계가 없는 데이터 3개를 선정하였다. sum_nVout, sum_nVin, sum_vsize는 2가지 상관분석 전부에 대해 높은 상관관계를 보이는 선정된 항목이다. 두 가지 상관분석을 적용하여 선정된 실험데이터는 모델의 성능 평가를 위해 학습데이터 80%, 검증데이터 10%, 실험데이터 10%의 비율로 구성하였다. 실험데이터 구성에 대한 정보는 표3과 같다.

표 3. 실험데이터 구성

블록의 높이(데이터 개수)	구분
100,000~180,000	학습데이터
180,001~190,000	검증데이터
190,001~200,000	실험데이터

Ⅲ. 실험 및 결과

본 장에서는 상관분석을 통해 선정된 LSTM 학습데이터를 이용하여 실험을 진행한다. LSTM 모델은 학습, 검증, 시험의 단계를 거친다. 학습단계는 같은 데이터를 여러 번 학습 하는 단계이다. 같은 데이터를 반복적으로 학습하게 되므로 해당 데이터에 맞는 예측 모델이 완성된다. 검증 단계는 학습에 사용되지 않은 데이터를 이용하여 모델의 성능을 검증하는 단계이다. 마지막 시험 단계는 학습 및 검증과정을 통해 완성된 모델의 성능을 평가하는 단계이다. 모델의 성능은 실제 값과 모델이 예측한 값의 평균 제곱 오차(Mean Square Error) 수치를 통해 평가한다. 평균 제곱 오차가 0에 가까울수록 모델의 성능이 좋다. 실험에 사용된 LSTM 모델의 정보는 표4와 같다.

표 4. 실험에 사용된 LSTM모델 정보

Hyper-Parameter	Method / Value
Data_Normalization	Min-Max Scaler
Sequence_Length	5
Number of Hidden_Unit	1
Loss_Function	MSE(Mean Square Error)
Optimizer	Adam

실험데이터 항목 간의 편차를 줄이기 위해 0과 1 사이의 값으로 정규화를 해주는 Min-Max Scaler 기법을 적용하였고, Sequence_Length는 5로 설정하였다. 실험데이터의 성능 비교를 극대화하기 위해 Hidden_Unit의

개수는 1로 설정하였다. Loss_Function은 모델의 성능을 객관적으로 비교하기 위해 평균 제곱 오차를 적용하였고, Optimizer는 기계학습에 보편적으로 쓰이는 Adam Optimizer를 적용하였다.

실험데이터를 학습하여 나온 모델의 실험 결과는 그림 3, 4와 같다. 피어슨 상관계수 분석을 적용하여 선정된 데이터를 학습한 LSTM 모델의 평균 제곱 오차는 피어슨 상관계수의 값이 클수록 작게 나왔다. 피어슨 상관계수가 가장 큰 데이터 항목인 sum_nVin은 LSTM 모델의 평균 제곱 오차가 가장 작게 나왔다. 따라서, 피어슨 상관계수 분석을 적용하여 나온 상관관계가 있는 데이터는 상관관계가 없는 데이터보다 LSTM 모델의 학습에 적합한 데이터라고 판단할 수 있다.

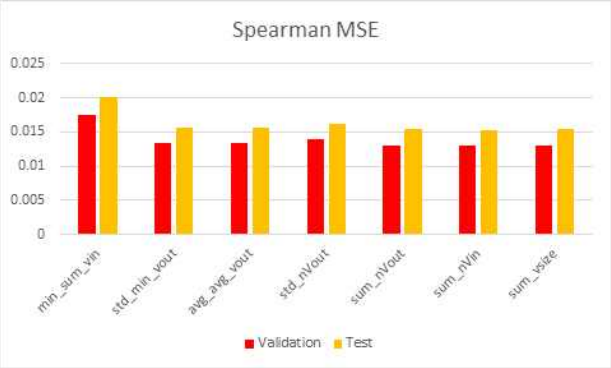


그림 3. 스피어만 상관계수 분석을 적용한 실험 결과

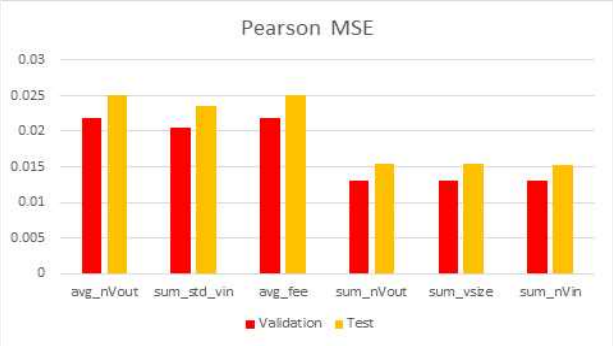


그림 4. 피어슨 상관계수 분석을 적용한 실험 결과

스피어만 상관계수 분석의 경우, 전반적으로 상관관계가 없는 데이터보다. 상관관계가 있는 데이터를 학습한 LSTM 모델의 평균 제곱 오차가 낮게 나왔지만, 스피어만 상관계수 값이 가장 큰 데이터 항목인 sum_vsize는 sum_nVin 보다 평균 제곱 오차가 높게 나왔다. 피어슨 상관계수 분석과 비슷한 데이터 항목이 선택되었으나 스피어만 상관계수와 LSTM 모델의 평균 제곱 오차의 관계는 불규칙적이다. 2가지의 상관분석을 적용하여 실험한 결과 피어슨 상관계수 분석이 비트코인 트랜잭션 수를 예측하는 LSTM 모델의 학습데이터를 찾는 방법으로 조금 더 적합하다.

Ⅳ. 결론 및 향후 연구

본 논문은 비트코인 네트워크에 있어 중요한 항목인 비트코인 블록의 트랜잭션 수를 예측하기 위한 LSTM 모델의 학습데이터 선택기법을 제안하였다. 제안하는 방법은 실험을 통해서 타당성을 검증하였다. 향후 연구로는 본 논문에서 언급한 2가지 상관관계 분석 이외의 다양한 통계분석을 적용하여, 올바른 학습데이터를 선정한 뒤, 비트코인 블록의 트랜잭션 수를 정밀하게 예측하는 LSTM 모델을 설계할 계획이다.

참 고 문 헌

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

[2] Bianconi, Gabriel, and Mahesh Agrawal. "Predicting Bitcoin Transactions with Network Analysis." (2017).

[3] Gers, Felix A., Jürgen Schmidhuber, and Fred Cummins. "Learning to forget: Continual prediction with LSTM." (1999): 850-855.

[4] Benesty, Jacob, et al. "Pearson correlation coefficient." Noise reduction in speech processing. Springer, Berlin, Heidelberg, 2009. 1-4.

[5] Zar, Jerrold H. "Significance testing of the Spearman rank correlation coefficient." Journal of the American Statistical Association 67.339 (1972): 578-580.

비트코인 데이터의 분석 및 시각화를 위한 웹 서버 구축

신혜영, 김대용, 이기영, 주홍택
계명대학교 컴퓨터공학과

oe_daphnea@naver.com, imdy1207@gmail.com, lkydig@naver.com, hongtaek.ju@gmail.com

Web Server for the Analysis and Visualization of Bitcoin Data

Hye-yeoung Shin, Daeyong Kim, Keeyoung Lee, Hongtaek Ju
Dept. of Computer Engineering, Keimyung University

요 약

비트코인(Bitcoin)의 노드에서 블록과 트랜잭션 데이터를 수집하고 분석하여 제공해주는 웹사이트는 현재 많이 있다. 그러나 이들이 제공하는 정보는 단편적이어서 심도 깊은 분석과 비트코인 블록체인 연구에 사용하기에는 한계가 있다. 비트코인 플랫폼의 성능 향상 혹은 기능 추가를 유추해내기 위해서는 종합적인 데이터를 수집하고 분석하여 그 결과를 시각화 하여 표현해야 할 필요가 있다. 본 논문에서는 복수의 노드에서 데이터를 수집하여 분석하고 이를 시각화 하는 모니터링 시스템의 웹 서버 구축 방안을 제시한다.

Ⅰ. 서 론

비트코인은 2008 년 나카모토 사토시(Nakamoto Satoshi)란 인물이 발표한 'Bitcoin: A Peer-to-Peer Electronic Cash System'[1]논문을 바탕으로 2009 년 개발되었다. 비트코인은 퍼블릭(public) 블록체인으로 누구나 제한 없이 자유롭게 참여할 수 있으며, 참여자(peer/node)는 블록체인에 기록된 모든 정보를 공유한다.

비트코인 노드에서 수집된 데이터(노드 정보, 블록 정보 등)를 바탕으로 분석 서비스를 제공해주는 대표적인 웹 사이트가 아래와 같이 있다.

	①	②	③	④	⑤
Nodes (Peers)	X	○	X	X	X
Block Size	○	○	○	○	○
P2P Messages	X	○	X	X	X
Hash	○	X	X	X	X
Transaction	○	○	○	○	○
Mempool Size	○	○	X	X	X
Difficulty	○	○	○	X	X
Mining Information	X	○	○	X	○
Tx Data	○	○	X	X	X
Tx Detail	○	X	X	X	X
Tx/Fee	○	○	X	X	X

① blockchain.com ② statoshi.info ③ btc.com
④ blockexplorer.com ⑤ tradeblock.com

[표 1] 대표적인 분석 웹 사이트

위 웹사이트에서 제공하는 분석 정보는 한 노드에서 수집된 트랜잭션과 블록 데이터를 분석하여 보여주고 있다. 이 정보로 비트코인에 대하여 단순한 이해는 가능하겠지만, 이를 이용해 비정상 참여자를 모니터링 하거나, 이상 거래 발생을 확인하는 등 다양하고 심도 깊은 분석과 연구가 불가능하다.

성능 향상 혹은 필요한 기능 도출을 위한 서비스 제공을 위해서는 여러 개 노드에서 데이터를 수집하고 트랜잭션 및 블록 데이터 이외에도 노드의 시스템 정보, 네트워크 정보 등 더 종합적인 데이터를 수집하여 분석한 결과를 시각적으로 제공하여야 한다[2]. 이를 위해 본 논문에서는 여러 개 노드에서 데이터를 수집하여 분석된 비트코인 데이터를 시각화 하는 모니터링 시스템 구축 방안을 제시한다. 본 논문에서 제안한 방법으로 수집된 데이터와 분석 결과는 비트코인 플랫폼의 성능을 개선하기 위한 근거자료로 활용되며 앞으로 비트코인 플랫폼에 추가되어야 할 새로운 기능에 대한 논의의 단초가 될 수 있다.

Ⅱ. 관련연구

비트코인 플랫폼에는 블록 ID 등을 포함한 블록에 관한 데이터, 트랜잭션 ID 와 입출력과 같은 트랜잭션 데이터, 코인의 주소와 같은 지갑에 관한 데이터, 트랜잭션의 유효성을 인증하는 암호에 관한 데이터 등 수 많은 데이터들이 있다. 이러한 데이터들을 활용한 분석을 통하여 더 많은 정보들을 가공해 낼 수 있다. 실제로 그러한 데이터들을 분석하여 DDoS, Ransomware 에 사용된 거래를 탐지하여 범죄의 피해를 예방하고, 방지하는 사례가 있다. 이와 같이 많은 비트코인 데이터들을 좀 더 쉽게 이해하기 위해서 통계분석을 포함하여 여러가지 분석을 실행하는 비트코인 모니터링 시스템에 관한 연구들이 수행되었다

.

name	Visualization Method		
	Approach	Visualization	Programming / Library
Bit bonkers	Network structure is not clearly visible	Cube → Last block from blockchain. The size in kilobytes → The size of the block its.	Three.js, Oimo.js
Bitcoin City	Distribution of nodes around the globe	Bitcoin node	Python
Binodes	City as Bitcoin transaction	City as transaction, road as blockchain	Isomer.js
Block seer	Node as link to transaction	Detailed tree diagram	-
Daily blockchain	Network of Blockchain, Realtime transaction, evolving hubs.	Evolving Hub	Vivagraph.js
Elliptic	History of Blockchain since the beginning of creation	Node → transaction	-
Interact	Live Bitcoin transaction	Size of the node → Volume of transaction Node → Link to transaction	-
Live globe	Transaction and latest discovery	Latest discovered Blockchain blocks	WebGL

[표 2] 블록체인 모니터링 시스템 서비스 별 개발 방법

KNOM Conf.2018 에서 소개된 '블록체인 네트워크 모니터링 및 분석시스템' [3] 에서는 블록체인 네트워크 모니터링을 통해 블록체인 데이터들을 분석하여 불법적인 거래를 탐지하는 모니터링 시스템의 방향을 제시하였다. 해당 논문에서 설계한 블록체인 모니터링 시스템은 Web Server 를 통해 사용자들에게 시각화된 데이터를 보여준다. 본 논문은 위 논문의 Web Server 기능을 확장한 결과를 제시한다.

H. Kazuno 등은 블록체인의 데이터들을 모니터링 하여 실시간으로 시각화하고 불법적인 거래를 추적하고 분석하는 방안을 제시하였다[4]. 주소 통계와 그래프로 나타낸 트랜잭션의 관계 및 경로를 발견하고, 이미 알려진 주소의 클러스터링 등의 데이터를 조합하여 분석한다. 분석의 결과로 비트코인 시장과 Ransomware, DDoS 를 통해 불법적으로 강탈한 3 가지의 범죄 사례에 적용하여 시스템의 유용성을 검증하였다.

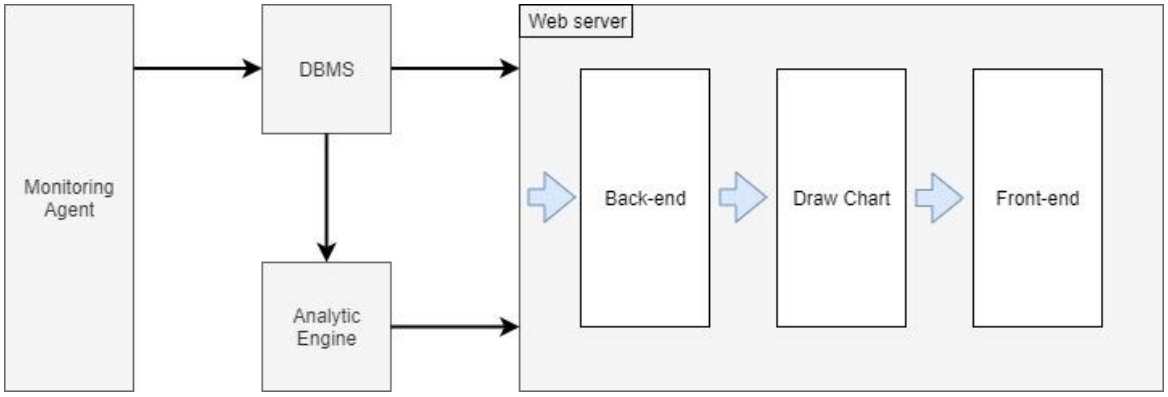
Ideva 등은 블록체인의 거래를 시각화 한 여러 개의 결과물들을 체계적으로 검토하고 평가한 결과를 제시하고 있다[5]. 검토하고 분석한 모니터링 시스템은 1. Bit bonkers, 2. Bitcoin City, 3. Binodes, 4. Block seer, 5.Daily blockchain, 6. Elliptic, 7. Interact, and 8. Live globe 이며 그 결과는 표 1 과 같다. 해당 연구를 통해 블록체인 데이터를 시각화 하기 위해 어떤 그래프가 사용되었고, 어떠한 것들을 이용하여 구축되었는지 분석하였다.

비트코인의 네트워크 분석과 관련된 연구로 'Exploring the Bitcoin Network' [6]은 비트코인 트랜잭션의 최근 데이터로 만들어진 그래프를 바탕으로, 그래프 마이닝 알고리즘을 이용하여, 네트워크 분석과 통계를 서술하고 있다. 비트코인의 많은 데이터들과 현재 네트워크의 상태를 표와 그래프로 나타내어 분석하였다. 또한 이러한 데이터를 바탕으로 하여, 데이터들의 추이에 따른 블록체인에서 발생하는 현상들을 설명하였다.

위에서 소개한 다양한 연구들을 통해 비트코인 모니터링 통해 분석한 블록체인의 데이터들을 바탕으로 다양한 결과들을 예기할 수 있으며, 그에 따른 문제점을 파악하여 비트코인 시스템의 구성 및 개발뿐만 아니라 운영, 유지 및 네트워크 자원 관리 등에 기여하여, 더욱 신뢰성 있는 블록체인 시스템을 구축할 수 있다. 따라서 본 연구에서는 가장 거래가 활발한 블록체인 플랫폼인 비트코인의 데이터 분석 및 시각화를 위한 블록체인 모니터링 시스템을 설계하고, 구현하는 방안을 제시한다.

Ⅲ. 설계

비트코인 데이터의 분석 및 시각화를 위한 모니터링 전체 시스템은 <그림 1>과 같다. 실제 비트코인 네트워크의 풀 노드에 모니터링 에이전트를 구동해 실시간으로 데이터 수집한다. 수집된 데이터는 데이터베이스에 저장된다. 데이터베이스에 저장된 데이터는 웹 모듈에서 원천 데이터를 그대로 사용하여 그래프나 원본 데이터를 보여주며 분석 도구를 사용한 분석 엔진에서 데이터를 분석하고 그 분석 결과를 웹 서버에 나타낸다. 웹 서버에서는 데이터베이스에서 가져온 데이터와 분석된 데이터를 백엔드에서 가공한다. 백엔드는 데이터베이스나 분석서버에 요청을 하여

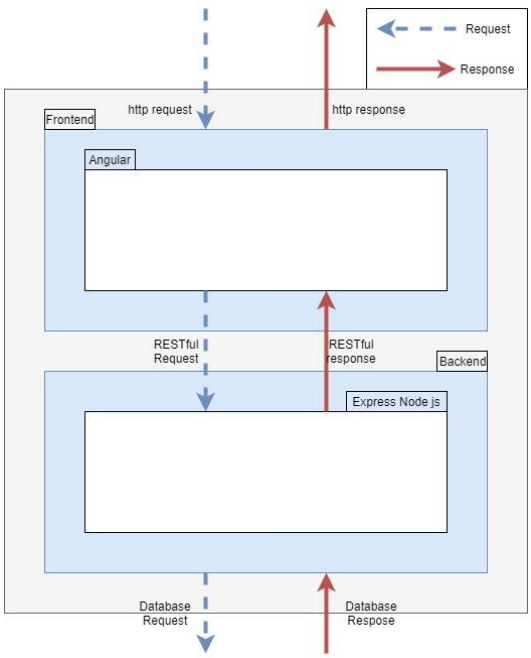


<그림 1> 블록체인 모니터링 시스템 전체 구조

데이터를 가져온다. 프론트엔드에서 높은 가시성을 위해 한번 더 가공하여 그래프나 표로 만들어 진다. 분석 데이터의 요청에 따른 결과는 그래프를 그리기 위한 데이터형식으로 가공하며, 블록체인의 가공되지 않은 데이터는 검색 결과로써 보여주기 위해 표의 형식으로 표현 되도록 가공 한다. 차트는 가공되어 있는 데이터를 RESTful API 형식으로 데이터를 요청하여 결과값을 받아 차트로 표현한다. 데이터의 요청은 특정시간마다 다시 요청하여 그래프 갱신을 통해 사용자에게 실시간 데이터로 보이게 한다. 프론트엔드는 사용자 편의를 위해 시각화를 지원하며 원하는 데이터를 표시하도록 한다.

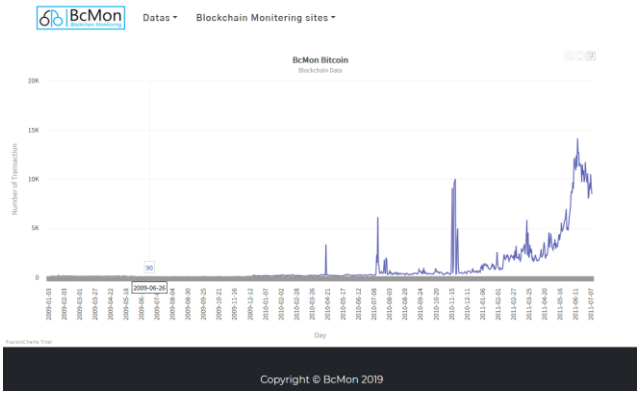
Ⅳ. 구현

본 연구에서 설계된 시스템은 높은 이식성과 개발시간 단축 그리고 관리의 편리성을 높이기 위해 도커 컨테이너를 사용하여 각 모듈을 마이크로 서비스 아키텍쳐 방식으로 구현 하였다. 비트코인 데이터 수집을 위한 에이전트 모듈은 풀 노드로부터 수집되는 데이터를 받아 전처리 과정을 통해 데이터베이스에 저장할 수 있는 형식으로 변형 되고, 최종적으로 데이터베이스에 저장된다. 데이터베이스는 MongoDB 를 사용하여 구현하였다. 분석 모듈에서 블록체인의 데이터를 날짜별로 정리하여 그래프로 출력하였다. 블록체인 데이터의 시각화를 위한 웹 모듈은 그림 2 와 같이 구현하였다. 본 시스템의 백엔드는 Express Node js 를 사용하여 구성되었다. 백엔드는 Clients 의 요청을 받으면 데이터베이스로부터 데이터를 가져와 그 결과를 RESTful API 형식으로 표현하고, 이 과정에서 그래프와 표에 들어갈 데이터로 분리되어 각각의 데이터들을 가공한다. 이후, 프론트엔드와 그래프를 작성하는 모듈로 데이터를 전달한다. 프론트엔드는 Angular 와 그래프를 그리기 위한 Fusioncharts 를 사용하여 시각화 모듈을 제작하였다. Fusioncharts 는 웹과 모바일 앱을 위한 Javascript 기반의 솔루션으로써, 다양한 그래프, 맵, 대쉬보드 등을 제공한다. 프론트엔드에서는 사용자의 요청에 따른 데이터를 시각화 하는 기능을 한다.



<그림 2> 블록체인 모니터링 데이터 시각화 시스템 구조

구현된 시스템은 <그림 3>에서 보여진다. 상단 바에 Datas 라는 메뉴를 클릭하면 블록체인에 관해 분석된 자료들 리스트가 나오며 확인하고 싶은 목록을 선택할 시에는 그래프가 그려진다. 그래프의 데이터는 5 분 단위로 재요청하여 갱신된다. 그래프는 확대, 축소와 이동 등의 기능을 가지고 있다. <그림 3>의 그래프는 일간 블록 크기의 평균 데이터를 출력한 것이다. 현재까지의 모든 데이터 뿐만 아니라, 확대와 축소 기능을 이용하여 선택한 기간의 데이터만을 모니터링 할 수 있다.



<그림 3> 블록체인 모니터링 시스템 웹 차트 표현

Ⅴ. 결과

본 논문에서는 모니터링 에이전트가 비트코인의 풀 노드에서 추출한 데이터를 데이터베이스에 저장하고, 저장된 데이터를 실시간으로 분석하여 그 결과를 시각화 한 다음, 웹을 통하여 모니터링 가능한 비트코인 실시간 분석 시스템의 웹 서버 설계 및 구현 방법을 제시한다.

본 연구에서는 비트코인의 데이터 수집, 분석 및 시각화를 연구하였지만, 향후 이더리움, 하이퍼레저 등과 같은 다양한 블록체인 플랫폼들에 적용되는 시스템으로 확장시킬 예정이다. 본 연구를 통해 개발된 시스템은 블록체인 네트워크 모니터링을 통한 성능 개선, 블록체인 서비스 개발, 비정상적 사용의 탐지나 보안적인 문제를 분석하고 추적하는 등에 활용이 가능할 것으로 기대된다.

ACKNOWLEDGMENT

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

참 고 문 헌

[1] Nakamoto, Satoshi. “Bitcoin: A peer-to-peer Electronic Cash System”(2008)

[2] Anton Badev and Matthew Chen, “Bitcoin: Technical Background and Data Analysis”(2014)

[3] 고경찬, 정태열, 유재형, 홍원기,”블록체인 네트워크 모니터링 및 분석시스템”, KNOM Conf (2018)

[4] Hiroki Kuzuno, Christian Karamyz, “Blockchain Explorer: An Analytical Process and Investigation Environment for Bitcoin”(2017)

[5] Tri A. Sundara, Ideva Gaputra, Siska Aulia, “Study on Blockchain Visualization”, INTER NATIONAL JOURNAL ON INFORMATICS VISUALIZATION, (2017)

[6] Exploring the Bitcoin Network, WEBIST 2014 - International Conference on Web Information Systems and Technologies(2014)

IoT 플랫폼을 위한 MQTT 클러스터에서 서브스크라이버 배정 방안

강귀영, 석승준*
경남대학교 컴퓨터공학과
randolph12@nate.com, *sjseok@kyungnam.ac.kr

Subscriber Assignment Method in MQTT Cluster for IoT platform

Gwi-Yeong Kang, Seung-Joon Seok*
Dept. of Computer Engineering, Kyungnam Univ.

요 약

본 논문에서는 Publish/Subscribe 시스템 중 하나인 MQTT 프로토콜의 브로커를 분산하고 분산 브로커 환경에서 동적으로 추가되는 퍼블리셔와 서브스크라이버를 분산된 브로커에게 배정하는 알고리즘을 제안한다. Publish/Subscribe 시스템에서 중앙 집중식의 브로커는 연결과 메시지의 손실이 발생 할 수 있다. 이러한 문제를 해결하기 위해 분산 브로커 환경이 제안되었으며 이러한 환경에서 클라이언트의 배정 방법에 대해서 논의되고 있다. 본 논문에서는 브로커의 부하와 메시지 교환 중 발생되는 부하와 통신비용을 고려하여 토픽을 공유하고 통신비용을 기준으로 클라이언트를 배정하는 알고리즘을 제안하였다. 제한하는 알고리즘은 실험을 통해 기존의 방법보다 시스템 부하와 통신비용이 감소하는 것을 확인 할 수 있었다.

Ⅰ. 서 론

IoT(Internet of Things)는 표준화된 통신 프로토콜을 사용하여 대상물들 사이에서 상호 연결되어 정보를 수집하고 가공을 통해 다양한 서비스를 제공하는 기술이다. IBM은 2012년 20억의 인구가 스마트폰 및 스마트 디바이스를 통해 연결되어 있으며 2020년에는 사물까지 포함된 500억개가 연결되는 거대한 네트워크가 형성 될 것이라 예측하고 있으며[1] 이러한 환경에서 IoT란, 모든 사물이 스스로 정보를 생성하고 공유하여 다수의 사용자와 개발자가 정당한 절차를 거쳐 쉽고 편리하게 서비스에 접근할 수 있는 진화된 형태이다.

IoT 디바이스는 제한된 네트워크 환경과 한정된 장비로 동작하기 때문에 HTTP(HyperText Transfer Protocl) 같은 웹 프로토콜의 사용보다 저사양 센서 및 IoT 디바이스에 적용할 수 있는 AMQP(Advanced Message Queuing Protocol), MQTT(Message Queue Telemetry Transport), CoAP(Constrained Application Protocol) 등과 같은 낮은 전력을 필요로 하고 각각의 클라이언트 사이에 비동기적으로 작동하는 가벼운 통신 프로토콜인 Publish / Subscribe 시스템이 필요하다.

개방형 IoT 플랫폼이 제공하는 다양한 형태의 서비스들 중 다중의 지역을 범위로 광범위한 IoT 서비스 지역에 동적으로 증가하고 감소되는 많은 수의 클라이언트 연결과 클라이언트에서 발생되는 메시지를 중앙 집중식의 브로커만으로 관리하기는 어렵다. 브로커를 지역마다 설치하여 발행되는 메시지를 관리하는 분산된 브로커 시스템이 필요하다. 그러나 브로커가 분산 관리 될 경우 브로커 사이의 메시지 교환이나 서비스하는 토픽의 메시지의 전달을 위한 라우팅 경로와 어느 브로커에 클라이언트가 연결될 것인지 고려해야할 요소들이 발생하게 된다. 이러한 문제를 해결하기 위해 브로커들로 구축 하는 오버레이 네트워크 방식과 P2P 방식의 메시지 라우팅, 그리고 클라우드 비용과 클라이언트와 브로커 사이의 대기시간을 기준으로 배정 문제를 해결하려는 연구들이 선행되어 왔다.

본 논문에서는 넓은 지역에서의 IoT 서비스 지원을 고려하여 Publish / Subscribe 시스템 중 하나인 MQTT 분산 브로커 환경에서, 동적으로 추가되는 서브스크라이버를 분산된 브로커 중 하나에 배정하는 방법을 연구한다. 분산된 브로커에서 수집한 부하를 기준으로 서브스크라이버의 위치, 브로커의 위치를 통해 각 브로커의 부하와 통신비용을 고려하여 IoT 플랫폼에서 분산된 브로커들의 부하가 적절히 분산되어 전체 시스템의 부하와 통신비용을 줄이기 위해 분산된 브로커에서 토픽을 공유하고 클라이언트를 가능한 통신비용이 적은 브로커에게 배정하는 알고리즘을 제안한다. 실험을 통해 기존의 다른 방식의 알고리즘보다 브로커의 부하와 통신비용, 시스템 전체의 부하와 통신비용이 감소하는 것을 확인 할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 방법의 연구 배경과 분산 브로커 환경에서의 문제를 다루는 관련 연구들에 대하여 기술한다. 그리고 3장에서는 고려하는 분산 브로커 시스템의 개요를 기술한다. 4장은 부하와 통신비용을 고려한 배정 알고리즘을 제안하고 5장에서는 실험 환경과 제안하는 방법을 통한 실험 결과를 나타낸다. 마지막 절에서는 결론을 맺는다.

Ⅱ. 연구 배경

a. 개방형 IoT(Internet of Things)

IoT 서비스의 주체들은 자신의 플랫폼을 IoT 표준하기 위해 별도의 플랫폼을 서비스하고 있다. 이는 폐쇄적이고 수직적인 플랫폼으로 설치를 위한 시간과 비용이 증가하며 서비스 운용을 위한 유지보수 비용이 크다. 사용자와 어플리케이션 개발자는 플랫폼에 접근이 제한되거나 불가능해지며 또한 서비스의 융합을 제공하기 위해 플랫폼 간 연동에도 큰 문제를 가지게 된다.

개방형 사물인터넷은 사용자들이 절차를 거쳐 쉽고 편리하게 서비스를 이용할 수 있는가를 나타내는 개념이다. 개방형 플랫폼은 사물과 서비스가 요구하는 공통기능을 제공하여 사업자들이 서비스를 생산, 관리할 수 있고 그 서비스를 사용 할 고객이 플랫폼을 이용하거나 개발하는 것에 대한

편의가 제공되어한다. 사용자가 서비스와 플랫폼에 효율적으로 접근 할 수 있는 관리 구조와 서비스 구조가 필요하고 개발자들이 필요로 하는 기능을 사용하기 쉽게 제공해야 한다. 적은 비용으로 다양한 서비스를 만들 수 있도록 편의를 제공해야 한다. 사물과 서비스에 독립적으로 동작하여 다양한 디바이스 수용이 가능하며 유지비용이 저렴하고, 공통 인터페이스 활용으로 서비스 간 융합 및 연계가 용이해진다.

b. 분산 브로커

브로커는 퍼블리셔가 발행한 모든 메시지를 수집하고 필터링 및 전달을 해당 서브스크라이버에게 배포한다. IoT 플랫폼이나 SNS와 같은 응용 서비스에서 중앙 집중식 브로커 모델이 사용된다. 중앙 집중식 브로커 모델은 개방형 IoT 플랫폼에서 IoT 데이터로 문제를 일으킬 수 있다. IoT 디바이스에 의해 자동적이고 지속적으로 생성되는 엄청난 양의 실시간 데이터가 중앙 집중식의 브로커에 집중되면 네트워크 대역폭을 압박하고 부하가 가중되어 브로커에 도착하는 데이터가 손실 되거나 클라이언트들의 연결이 끊어질 수 있다. 그리고 IoT 디바이스는 지역적으로 분포되어 있으며 각 장치에서 생성되는 데이터는 생성된 지역에서 활용 되는 데이터 일 가능성이 높다. 지역에서 생성되는 데이터가 이벤트 중심 서비스에서 사용되는 경우, 실시간으로 데이터를 서브스크라이버와 교환해야 하기 때문에 브로커가 여러 지역으로 분산되어야한다. 분산 브로커는 퍼블리셔로부터 발행되는 메시지를 수신하고 브로커는 상호 통신이 되어야 하며 해당 데이터를 원하는 서브스크라이버에게 메시지를 배포한다.

c. 클라이언트의 분산 브로커 배정

현재 분산 브로커에서는 메시지의 라우팅과 클라이언트의 배정에 관한 이슈가 다양하게 연구되고 있다. 각 브로커 별로 해당하는 토픽을 서비스하는 브로커와 토픽을 서비스하지 않는 브로커가 퍼블리셔로 부터 발행되는 메시지를 최적의 경로를 통해 토픽을 구독 중인 서브스크라이버가 있는 브로커에게 배포하기 위한 메시지 라우팅 방법[2-4]은 기존의 네트워크 구조에서 확장 할 수 없으며 불필요한 메시지의 전달로 인해 전달 통신 비용 및 컴퓨팅 자원이 소모된다. 이러한 점을 해결하기 위해 선행되어온 연구는 물리적 연결을 모두 가상화하여 오버레이 네트워크로 만들거나 SDN(Software defined networking) 컨트롤러를 이용하여 메시지를 최적의 경로로 라우팅 되도록 제안 하였다.

분산된 브로커에 클라이언트를 최적의 방법으로 배정 시켜주는 방법[5-6]은 분산된 브로커에 클라이언트를 배정하는 방법으로 브로커의 부하를 고려해야 할 뿐만이 아니라 클라이언트와 서버간의 통신 대기시간, 사용자와 사용자간의 통신 속도 및 대기 시간도 고려해야 한다. 클라이언트와의 가까운 위치 배정, 네트워크 상황을 고려한 배정, 무작위한 브로커 배정 등의 방법들이 제안 되고 있으며, QoS를 보장 할 것인지, 브로커를 가능한 낮은 비용으로 서비스를 제공 할 것인지, 아니면 두 가지의 입장을 모두 고려한 방안을 제시 할 것인지 각각의 해결 방안에 따라 제시하였다.

Ⅲ. 고려하는 IoT 분산 브로커 시스템 개요

본 논문에서는 개방형 IoT 플랫폼 기반의 분산 브로커 시스템에서 클라이언트인 서브스크라이버가 분산된 브로커 중 하나에 연결될 수 있도록 클라이언트를 브로커에 배정하는 알고리즘을 제안한다. 제안하는 분산 브로커 환경은 클라이언트, 브로커, 로컬 브로커, 로컬 클러스터, 인접 클러스터, 컨트롤러로 구성 되어 있다. 클라이언트는 퍼블리셔와 서브스크라이버로 구성되어 있다. 클라이언트와 가장 가까이에 있는 브로커를 로컬 브로커라고 하며 클라이언트와 통신비용은 가장 작다. 클라이언트는 로컬 브로커의 정보를 알고 있어서 다른 정보가 필요 없이 브로커로 연결 신청이 가능하다. 로컬 클러스터는 클라이언트와 인근의 브로커들의 집합으로서 클라이언트가 연결 될 때 인접 클러스터의 어느 브로커와 연결되어도 통신비용의 차이가 나지 않는다. 로컬 클러스터에 속해 있는 브로커 중에서 클라이언트의 토픽이 등록된 브로커 그룹을 G1, 클라이언트의 토픽이 등록되지 않은 브로커의 그룹을 G2로 명명한다. 로컬 클러스터와 가장 가까운 위치에 있는 브로커 집합을 인접 클러스터라고 명명하고 클라이언트와의 통신비용 증가가 크다. 컨트롤러는 제안 알고리즘을 수행하고 수행하기 위해 IoT의 모든 정보를 수집하고 분석한다.

Ⅳ. 다중 브로커 시스템에서의 서브스크라이버 배정 알고리즘

본 논문에서 제안하는 알고리즘은 토픽과 브로커를 공유하고 브로커의 부하가 높을 경우 통신비용에 손해를 보더라도 부하가 낮은 브로커에 배정하여 전체 시스템의 부하를 균등하게 유지한다.

a. 부하와 비용

비용은 부하비용과 통신비용으로 나뉜다. 각 브로커의 용량의 총 합으로 브로커의 용량은 브로커에 등록된 토픽 부하의 총 합이며 부하 비용이라고 한다. 토픽의 부하는 다음 식 (1)과 같다.

$$TOPIC\_LOAD = (Log_e X) + 1$$
$$X = Number\ of\ scriber(per\ Topic) \quad \text{식 (1)}$$

브로커는 수신한 메시지가 동일한 토픽일 때 메시지 버스를 통해 메시지를 전달하기 때문에 한 토픽을 구독하는 서브스크라이버가 많아질수록 토픽의 부하 증가율이 낮아진다. 한 토픽을 구독하는 서브스크라이버의 수가 증가 할수록 부하의 증가율이 감소하는 Log 그래프의 모습을 보인다.

브로커의 부하와 시스템의 부하는 각각 식 (2)와 같다.

$$BROKER\_LOAD = \sum_{t \in T_i} TOPIC\_LOAD(t)$$
$$SYSTEM\_LOAD = \sum_{b \in B_i} BROKER\_LOAD(b)$$
$$T_i = \text{브로커 } i\text{에 속한 토픽의 집합}$$
$$t = T_i\text{에 속한 토픽}$$
$$B_j = \text{클러스터 } j\text{에 속한 브로커의 집합}$$
$$b = B_j\text{에 속한 브로커} \quad \text{식 (2)}$$

통신비용을 나타내기 위한 파라미터는 <표 1>와 같이 정의한다.

〈표 1〉 배정 알고리즘의 통신비용 파라미터 정의

기호	파라미터 정의
C_0	서브스크라이버와 로컬 브로커의 통신비용
C_1	서브스크라이버와 로컬 클러스터의 통신비용
C_2	서브스크라이버와 인접 클러스터의 통신비용
B_1	동일 클러스터의 브로커 사이의 통신비용
B_2	인접 클러스터의 브로커 사이의 통신비용
P_0	퍼블리셔와 로컬 브로커의 통신비용

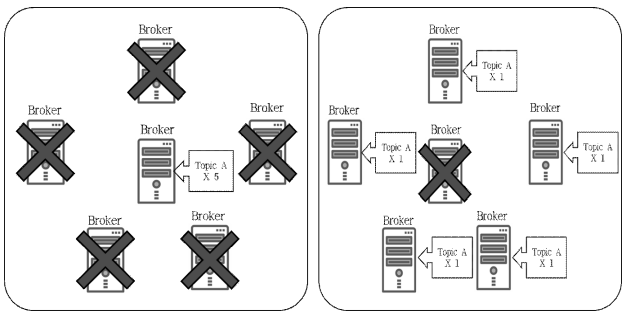
서브스크라이버의 통신비용은 로컬 브로커, 로컬 클러스터, 인접 클로스터의 브로커와의 연결 시 점차 증가하며 브로커 사이의 통신비용은 동일 클러스터의 브로커가 아닐 시 증가하게 된다.

$$NEWORK\_COST = \sum_{n=0}^{3} C_n + \sum_{n=1}^{2} B_n + \sum P_0 \quad \text{식 (3)}$$

위의 식으로 계산한 통신비용과 각 브로커의 부하비용을 균등하고 그 합을 낮게 유지하는 것이 제안하는 배정 알고리즘의 목표이다.

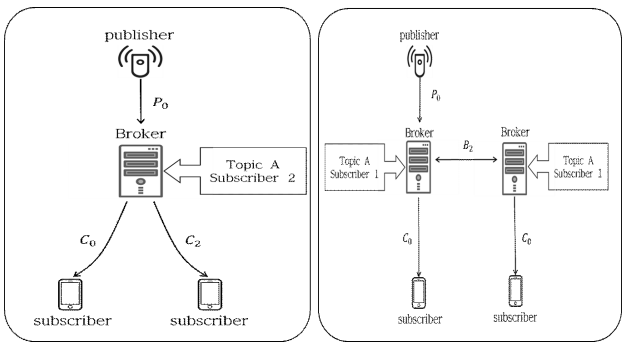
b. 알고리즘의 개념

제안하는 배정 알고리즘에서 부하와 통신비용을 낮추기 위한 방법으로는 클러스터 내의 브로커를 이용하며 토픽을 공유하는 것이다. <그림 1>과 <그림 2>에서는 로컬 브로커에 서브스크라이버가 구독할 토픽이 없다면 통신비용이 증가 하더라도 토픽과 브로커를 공유하는 것이 통신비용과 부하가 더 낮다는 것을 알 수 있다.



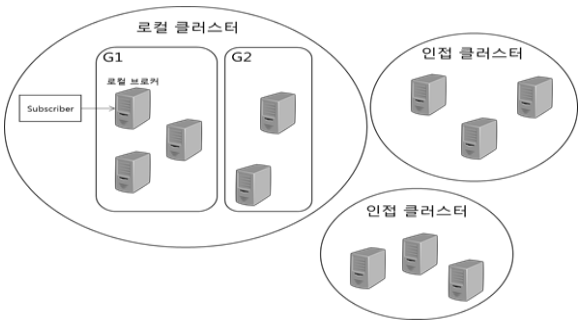
(a) 토픽을 공유하는 모델 (b) 토픽을 분배하는 모델

〈그림 1〉 분산 브로커 시스템의 부하



〈그림 2〉 분산 브로커 시스템의 통신비용

c. 제안하는 알고리즘



〈그림 3〉 고려하는 분산 브로커 환경의 구조

<그림 3>은 제안하는 MQTT 기반 분산 브로커 환경 구조를 나타내고 있다.

제안하는 알고리즘은 각 단계별로 설정된 임계값을 설정하여 설정 값 이하라면 현 단계에서 배정하고 초과 했을 때는 통신비용이 증가하더라도 현 단계의 브로커보다 유리한 성능을 낼 수 있는 브로커라고 판단을 내리고 부하가 낮은 다음 단계의 브로커에 연결을 시킨다. 클라이언트와의 위치를 기준으로 가장 가까운 위치의 브로커인 로컬 브로커, 통신비용이 크게 증가하지 않는 위치의 브로커들의 집합인 로컬 클러스터, 로컬 클러스터와 인접한 브로커들의 모임인 인접 클러스터로 나뉜다. 로컬 클러스터에 속해 있는 브로커 중 연결 신청한 클라이언트의 토픽이 등록된 그룹을 G1, 토픽이 등록되지 않은 그룹을 G2라고 한다. 제안하는 알고리즘은 로컬 브로커의 상태가 배정 받아도 좋은 상태인지를 설정된 조건에 따라 판단하여 클라이언트를 로컬 브로커에 배정함으로서 서비스에 있어서 통신비용을 최소화하기 위해 제안되었다. 만약 로컬 브로커에 클라이언트가 배정 받지 못하게 된다면 통신비용의 증가가 거의 없는 로컬 클러스터에 속해있는 브로커 중에서도 같은 토픽이 있는 G1의 브로커에 연결하여 브로커와 브로커 사이의 추가적인 통신비용을 최소화 한다. G1에서 배정 받지 못했을 경우 브로커 간의 통신비용이 발생하지만 G2의 같은 로컬 클러스터내의 브로커 중 용량이 낮은 브로커에 배정한다. 로컬 클러스터에 속한 브로커들의 부하가 모두 높다고 판단하여 배정 받지 못한 경우 인접 클러스터에 속한 브로커 중 부하가 낮은 브로커에 클라이언트를 배정한다. 만약 인접 클러스터에서도 브로커들의 부하가 높아 클라이언트가 배정 받지 못하여 전체적인 시스템의 부하가 높다고 판단되면 인접 클러스터 내의 같은 토픽 클라이언트들을 전부 로컬 클러스터의 가장 작은 브로커로 이전하여 전체적인 시스템의 부하의 감소를 기대한다.

V. 실험 및 결과 분석

〈표 2〉 시뮬레이터 환경과 실험 시나리오 변수

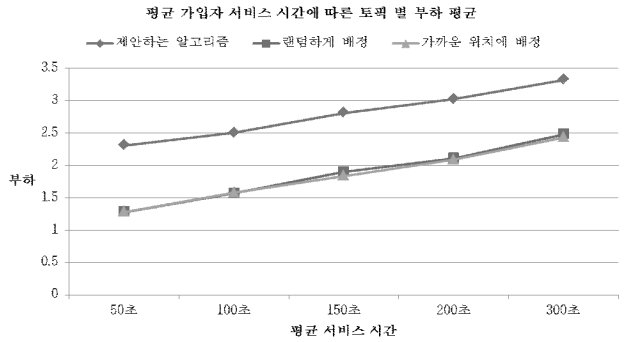
PC 사양	CPU : i7-4720 2.60GHz RAM : 16G
프로그래밍 언어	JAVA
클러스터 개수	5개
브로커 개수	클러스터 당 10개
토픽 개수	50개
서브스크라이버 생성 위치	2%의 확률로 브로커 중 한 곳에 위치
서브스크라이버 생성 주기	1초당 40개 생성
서브스크라이버 서비스 평균 시간	평균 100초

<표 2>는 실험을 위한 환경과 시나리오 변수이다. 실험은 크게 두 종류로 하나는 비교하기 위한 알고리즘 두 가지를 제시하고 제안하는 배정 알고리즘과의 성능비교를 하였고, 시나리오 환경을 변화하여 제안하는 알고리즘의 가장 적절한 임계값을 구하기 위한 실험을 하였다. 성능비교를 위한 실험은 서브스크라이버를 해시 값으로 추출하고 브로커의 개수로 나누어 나온 결과 값의 나머지에 해당하는 브로커에 브로커를 배정하는 랜덤 배정 알고리즘과 서브스크라이버가 생성될 때 가장 가까이에 있는 로컬 브로커에게 배정하는 가까운 위치에 배정하는 알고리즘, 부하와 통신비용을 고려한 제안하는 배정 알고리즘을 JAVA로 구현하여 시뮬레이터 환경과 실험 시나리오의 변수를 변화시켜가며 비교한다. 시나리오 변수 값을

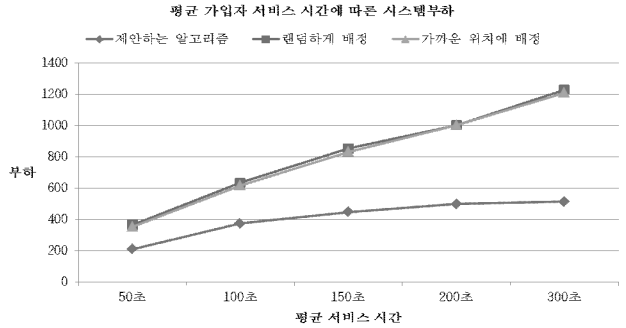
변화하여 최적 값을 구하는 실험은 기본적은 시뮬레이터 환경에서 환경의 변수를 다양한 입력 값으로 교체하여 제안하는 알고리즘의 최적의 값을 찾는다.

a. 실험결과

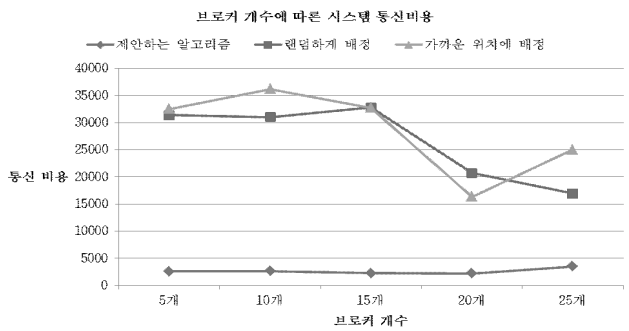
본 논문에서 두 가지 방법으로 실험을 진행한다. 하나는 제안하는 알고리즘의 성능을 비교하기 위해 시뮬레이터에 구현한 무작위 배정 알고리즘과 가까운 위치에 배정하는 알고리즘을 실행하고 제안하는 배정 알고리즘을 실행하여 나온 실험 결과 값으로 토픽 당 부하 평균, 시스템의 부하, 시스템의 통신비용으로 비교 후 정리한다. 브로커 개수, 평균 가입자 서비스 시간, 토픽의 최대 개수, 일부 브로커에 부하를 집중시키는 등 환경 변수를 변화하여 실험 결과를 비교한다. 다른 하나는 임계값의 위치 변화하여 각 임계 값 중 좋은 성능을 찾은 후 최고 성능 조합과 최저 성능 조합으로 실험하여 기본 성능과 비교한다. 또한 임계값을 조정하여 기본적인 모델, 서브스크라이버의 통신비용을 최소화하는 모델, 동일한 토픽에 배정하기 위한 모델을 비교하여 케이스 별 그래프로 나타내어 실험 결과를 비교한다.



〈그림 4〉 평균 가입자 서비스 시간에 따른 토픽 별 부하 평균



〈그림 5〉 평균 가입자 서비스 시간에 따른 시스템 부하



〈그림 6〉 브로커 개수에 따른 시스템 통신비용

실험을 통해 제안하는 서브스크라이버 배정 알고리즘이 다른 방식의 배정 알고리즘과 비교하여 좋은 성능을 나타내는 것을 확인하였고 서브스크라이버의 배정에 있어서 토픽의 공유가 증가할수록, 그리고 로컬 브로커나 로컬 클러스터의 브로커에 배정 될수록 시스템 전체의 부하와 통신비용이 감소하게 되는 것을 증명한다.

Ⅵ. 결론

본 논문에서는 분산 브로커 시스템에서 브로커의 부하와 통신비용을 고려하여 로컬 브로커, 로컬 클러스터, 인접 클러스터로 나누어 클라이언트를 브로커에 부하와 통신비용을 고려하여 배정하는 방법에 대하여 제안하였다. 제안하는 배정 알고리즘을 구현하기 위해 분산 브로커 시스템 환경을 시뮬레이터로 구현하였고 시뮬레이터에서 제안하는 알고리즘을 성능을 평가하기 위해 서브스크라이버가 무작위로 배정되는 알고리즘과 서브스크라이버의 위치와 가까운 위치에 배정되는 알고리즘을 구현하여 세 가지 알고리즘을 비교 분석 하였다. 또 실험 환경의 파라미터를 변경하여 가장 성능이 좋은 임계값을 찾기 위한 실험도 진행하였다. 세 가지 알고리즘을 비교한 결과 제안하는 알고리즘의 성능이 랜덤 배정 알고리즘과 가까운 위치에 배정하는 알고리즘 보다 2배 이상 낮은 부하와 월등히 낮은 통신비용을 나타내는 것을 확인 할 수 있다. 또한 제안하는 알고리즘의 토픽을 공유하고 통신비용이 낮은 브로커에 배정하는 방법이 분산 브로커 시스템에 있어서 부하를 낮추고 통신비용이 감소시킨다는 것을 실험으로 검증하였다. 향후 서브스크라이버의 둘 이상의 토픽 구독과 퍼블리셔의 여러 토픽으로 발행 할 때의 배정 알고리즘을 위한 추가 실험과 개선이 필요하다.

참 고 문 헌

[1] S. Hodges S. Taylor N. Villar J. Scott D. Bial P. and T. Fischer "Prototyping connected devices for the Internet of Things" Computer vol. 46 no. 2 pp. 26-34 Feb. 2013.

[2] R. Banno S. Takeuchi M. Takemoto, T. Kawano T. Kambayashi and M. Matsuo "Designing Overlay Networks for Handling Exhaust Data in a Distributed Topic-based Pub/Sub Architecture" Journal of Information Processing vol. 23 no. 2 2015.

[3] N. Carvalho F. Arajo and L. Rodrigues "Scalable QoS-based event routing in publish-subscribe systems" Technical report Feb. 2005.

[4] R. Banno J. Sun M. Fujita S. Takeuchi and K. shudo "Dissemination of edge-heavy data on heterogeneous MQTT brokers." In Cloud Networking (CloudNet), 2017 IEEE 6th International Conference on pp.1-7. 2017.

[5] Y. Teranishi T. Kawakami and Y. Ishi "A large-scale data collection scheme for distributed Topic-Based Pub/Sub" International Conference on Computing Networking and Communications (ICNC) vol. 1 pp. 235-241 Jan. 2017.

[6] Y Wang Y Zhang and J. Chen "An SDN-based publish/subscribe-enabled communication platform for IoT services [J]" China communications vol. 15 no. 1 pp. 95-106 2018.

End-to-end Network Resource Slicing in Mobile Edge Computing for 5G New Radio

Yan Kyaw Tun, Shashi Raj Pandey, and Choong Seon Hong

Department of Computer Science and Engineering, Kyung Hee University, South Korea

ykyawtun7@khu.ac.kr, shashiraj@khu.ac.kr, cshong@khu.ac.kr

Abstract

Wireless network slicing is one of the promising technologies for 5G new radio network to enhance the network capacity and the peak data rate of the current LTE network. Meanwhile, the upcoming technology so called mobile edge computing (MEC) offloading offloads the computation intensive task on the mobile device to the cloud server located at the edge of the cellular network. By offloading, MEC network can enhance the the computation capacities of the mobile devices and prolong the battery lives. However, there are still several challenges to address before deploying the aforementioned technologies in telecommunication industry. Among them, the efficient resource allocation is the most important issue. Therefore, in the work, we propose the energy efficient ene-to-end network slicing problem in the MEC network. Then, we deploy the block coordinate descent (BCD) approach to address the proposed problem. Simulation results show that our proposed scheme outperforms the existing schemes.

Keywords - End-to-end network slicing, mobile-edge computing(MEC), block coordinate descent (BCD).

1. Introduction

In order to address problems introduced by the exponential growth of the mobile devices with the emerging mobile applications (e.g., face recognition, virtual reality (VR), Augmented reality (AR) and so on), researchers in the telecommunication industry are proposing new technologies such as wireless network slicing, mobile-edge computing. Network slicing enables decoupling the current cellular network into two entities, such as infrastructure provider (InP) and mobile virtual network operators (MVNOs) where InP provides infrastructures and wireless network resources to the MVNOs and MVNOs provide different services (i.e., ultra-reliable and low latency communication (URLLC), enhanced mobile broad-band (eMBB), and massive machine type communication (mMTC)) to their mobile users [1, 2, 3]. Meanwhile, mobile-edge computing becomes the new and key technology for 5G new radio. Thereby, MEC can reduce the delay experienced by the mobile devices when compared with the mobile cloud computing, and fog computing [4, 5, 6]. However, the computation capacity (i.e., CPU) of the MEC is limited. Therefore, how to efficiently offload the computation intensive tasks of the mobile devices to the sever at the edge of the radio access network becomes important issue in the MEC network.

2. System Model and Problem Formulation

As shown in Fig. 1, an infrastructure provider (InP) deploys the macro base station (BS) integrated with the edge server. So, we will use the term edge server and base station interchangebaly. Then, the physical infrastucture is split into multiple virtual networks to support heterogeneous mobile services requests that are categorized into MEC service slice and traditional cellular service slice. Then, the InP will allocate each resource slice including cellular resource slice and MEC resource slice to each mobile virtual network operator. So, we will use the term slice and MVNO interchangeably throughout the paper. Mobile virtual network operators, i.e., service providers, are providing multiple services to their mobile users where each service has its own QoS requirements.

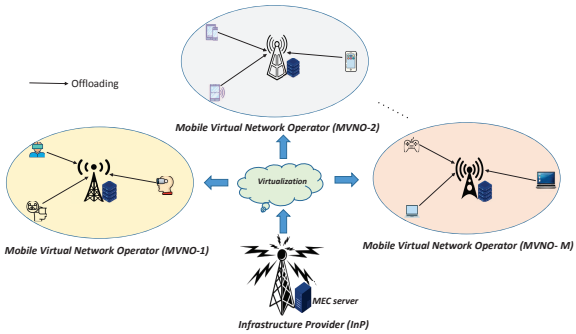


Figure 1: System Model

In this model, we assume that the base station is operating on the system bandwidth B and it is divided into the subchannels where each subchannel has the bandwidth ω. Moreover, an InP is providing the virtual networks/ resource slices (including communication resources such as sub-channel and transmit power, computation resource) to the M mobile virtual network operators/ service provider (SPs), denoted as a set $\mathcal{M} = \{1, 2, \ldots, M\}$. Then, MVNOs are providing mobile services to their mobile users and each MVNO has U mobile users, denoted as a set $\mathcal{U} = \{1, 2, \ldots, U\}$. Here we assume that each mobile user of each MVNO is generating a task a_{mu} to be executed. Then, the properties of a task of each mobile user in each MVNO can be represented as a tuple (d_{mu}, c_{mu}) where d_{mu} is the total data size (i.e., bits) of the task of user u of MVNO m, c_{mu} is the required CPU cycle to execute a bit of data. For each user of each MVNO, its computation task can be executed locally on mobile device or executed remotely at the MEC server. Therefore, let us introduce a binary variable x_{mu}, where $x_{mu} = 1$ indicates that the generated task is offloaded to the server, $x_{mu} = 0$ otherwise.

1) Local Computing: When the generated task of the user u of the MVNO m is executed locally, the latency to com-

plete the task execution can be formulated as follows:

$$t^L_{mu} = \frac{d_{mu}c_{mu}}{f^l_{mu}}, \forall u \in \mathcal{U}, \forall m \in \mathcal{M}, \quad (1)$$

where f^l_{mu} is the computation capacity of user u of the MVNO m. Moreover, the energy consumption of the user u can be expressed as follows:

$$E^L_{mu} = k(f^l_{mu})^2 c_{mu} d_{mu}, \forall u \in \mathcal{U}, \forall m \in \mathcal{M}, \quad (2)$$

where $k = 10^{-26}$ and it depends on the chip architecture of the mobile device. Then, the local computation overhead of the user u of MVNO m can be formulated as follows:

$$G^L_{mu} = \lambda_t t^L_{mu} + \lambda_e E^L_{mu}, \forall u \in \mathcal{U}, \forall m \in \mathcal{M}, \quad (3)$$

where λ_t and λ_e of the weighted parameters for local computation latency and local energy consumption.

2) Remote Computing: When the generated task of the user u of the MVNO m is executed remotely, the mobile user firstly transmits the task (i.e., input data) to the edge server. Therefore, the achievable data rate of the user u of the MVNO m is as follows:

$$R_{mu} = \omega \log_2(1 + \frac{p_{mu}h_{mu}}{N_0}), \forall u \in \mathcal{U}, \forall m \in \mathcal{M}, \quad (4)$$

where p_{mu} is the transmit power of the user u of MVNO m, h_{mu} is the achievable channel gain, and N_0 is the additive white Gaussian noise power. Then, we can formulate the transmission latency/delay experienced by the user u of MVNO m as follows:

$$t^O_{mu} = \frac{d_{mu}}{R_{mu}}, \forall u \in \mathcal{U}, \forall m \in \mathcal{M}. \quad (5)$$

Moreover, in this work, we assume that the computation capacity of the edge server is infinite. Therefore, the execution delay of the task of the user u in MVNO m can be ignored. Then, the energy consumption for the uplink transmission as follows:

$$E^O_{mu} = \frac{p_{mu}d_{mu}}{\omega \log_2(1 + \frac{p_{mu}h_{mu}}{N_0})}, \forall u \in \mathcal{U}, \forall m \in \mathcal{M}, \quad (6)$$

Finally, we can formulate the remote computation overhead experienced by the user u of MVNO m as follows:

$$G^O_{mu} = \lambda_t t^O_{mu} + \lambda_e E^O_{mu}, \forall u \in \mathcal{U}, \forall m \in \mathcal{M}. \quad (7)$$

In this work, the efficient generated task offloading and power allocation for the end-to-end network slicing in the MEC network is formulated as the optimization problem. The objective is to minimize the computation overhead experienced by the users of all MVNOs. Under the efficient offloading decision constraint, and power budget constraint of each mobile user, the optimization problem can be expressed as follows:

$$\min_{\mathbf{x},\mathbf{p}} \left(\sum_{m=1}^{M}\sum_{u=1}^{U} x_{mu}G^O_{mu} + \sum_{m=1}^{M}\sum_{u=1}^{U}(1-x_{mu})G^L_{mu} \right) \quad (8)$$

$$\text{s.t. C1} : x_{mu} \in \{0,1\}, \quad \forall u \in \mathcal{U}, \forall m \in \mathcal{M}, \quad (9)$$

$$\text{C2} : 0 \le p_{mu} \le P^{\mathbf{max}}_{mu}, \quad \forall u \in \mathcal{U}, \forall m \in \mathcal{M}, \quad (10)$$

where $P^{\mathbf{max}}_{mu}$ is the maximum transmit power of the user u of the MVNO m. Then, C1 represents the task offloading constraint, and C2 shows the power budget constraint of each mobile user in MVNOs. We can see that aforementioned optimization problem is the mixed integer and non-convex problem. Generally, it is difficult to solve. Therefore, we divide the original problem into two subproblems and each subproblem becomes convex. Then, we provide the close-form solution for each subproblem.

2.1 Task Offloading Problem (TOP)

In a given power allocation, the optimization problem expressed in the (8) can be transformed into the task offloading problem and it is as follows:

$$\min_{\mathbf{x}} \left(\sum_{m=1}^{M}\sum_{u=1}^{U} x_{mu}G^O_{mu} + \sum_{m=1}^{M}\sum_{u=1}^{U}(1-x_{mu})G^L_{mu} \right) \quad (11)$$

$$\text{s.t. C1} : x_{mu} \in [0,1], \quad \forall u \in \mathcal{U}, \forall m \in \mathcal{M} \quad (12)$$

where we firstly relax the offloading variable (i.e., binary variable) into the continuous form and the task offloading problem becomes convex problem. So, we can use the CVX solver to solve it.

2.2 Power Allocation Problem (PAP)

In a given task offloading, the optimization problem mentioned in (8) can be transformed into the power alloation problem and it is as follows:

$$\min_{\mathbf{p}} \left(\sum_{m=1}^{M}\sum_{u=1}^{U} x_{mu}G^O_{mu} + \sum_{m=1}^{M}\sum_{u=1}^{U}(1-x_{mu})G^L_{mu} \right) \quad (13)$$

$$\text{s.t. C2} : 0 \le p_{mu} \le P^{\mathbf{max}}_{mu}, \quad \forall u \in \mathcal{U}, \forall m \in \mathcal{M}, \quad (14)$$

where the above subproblem so called power allocation problem is also convex. Therefore, we can use the CVX solver to solve the aforementioned sub-problem in (13). However, we omit the proofs of convexity for two subproblems because of the limited space. We can get the solution of the optimization problem by solving the the above two subproblems alternatively.

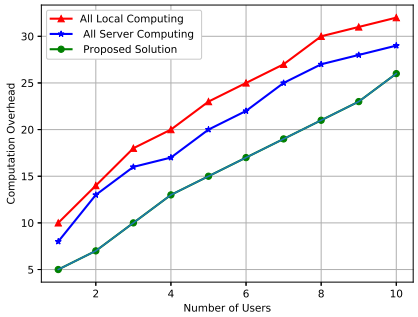


Figure 2: Tradeoff betwee the computation overhead and number of users

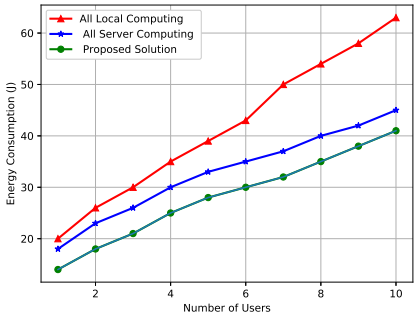


Figure 3: Tradeoff between energy consumption and number of users

3. Simulation Results

In our simulation section, we consider a single BS with the total system bandwidth 20MHz and the bandwidth of each subchannel is 180kHz. The additive white Gaussian noise power is $-174dBm/Hz$. The long distance path loss model in our simulation is $PL = 40\log_{10}(d_0) - 10\log_{10}(Gh_t^2h_r^2) + 10\gamma\log_{10}\frac{d}{d_0} + X_g$ where G is the gain product of transmitter and receiver, d_0 and d are the reference distance and actual distance between transmitter and receiver, h_t and h_r are heights of transmitter and receivers, and X_g is the random variable. Moreover, we consider the BS is serving 2 MVNOs and there are 5 mobile users in each MVNO. The maximum transmit power and the maximum computation (i.e., CPU) capacity of each mobile device is 1mW and 0.2GHz. The total input data size of each mobile device is 0.5MB and the required CPU capacity to execute one bit of input data is 500 cycles. In case of simplicity, in this simulation setting, we consider both λ_t and λ_e are 1. Fig. 2 shows the trade off between computation overhead and number of users. We can see that computation overhead depends on the number of users. When the number of users increases, the computation overhead also increases. Moreover, we compare our proposed algorithm with the all local computing and all server computing. It can be seen from Fig. 2 that our proposed solution outperforms other two schemes. The reason is that when each mobile user executes its computation task locally, more energy is consumed. Another one, the computation task offloaded to the edge-server has high latency. This is why, all local computing and all server computing is resulting in the high computation overhead.

Fig. 3 represents the trade-off between energy consumption and number of users of all MVNOs. From Fig. 3, we can see that the energy consumption is the highest when users of MVNOs execute their tasks locally (i.e., all local computing). Moreover, the energy consumption is higher than our proposed solution and lower than local computing when all users offload their computation tasks to the edge-server because of the energy consumption for uplink transmission.

4. Conclusion

In this work, we propose energy efficient computation task offloading and end-to-end network slicing scheme in the mobile edge computing (MEC). Then, we formulate the optimization problem as a minimizing computation overhead problem under the efficient offloading decision constraint, and power budget constraint of each mobile device. In the simulation section, we can see that our proposed solution outperforms other schemes. Moreover, our work is the very first work that consider both network slicing and mobile edge computing at the same time. In future, we will extend our work into multiple base stations (BSs) and multiple mobile-edge severs case.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (NRF-2017R1A2A2A05000995). *Dr. CS Hong is the corresponding author.

References

[1] M. Alsenwi, N. H. Tran, M. Bennis, A. K. Bairagi, and C. S. Hong, "eMBB-URLLC resource slicing: A risk-sensitive approach," *IEEE Communications Letters*, vol. 23, no. 4, pp. 740–743, apr 2019.

[2] M. A. C. W. Z. C. S. H. Yan Kyaw Tun, Shashi Raj Pandey, "Weighted proportional allocation based power allocation in wireless network virtualization for future wireless networks," in *The 33rd International Conference on Information Networking (ICOIN 2019), IEEE*, 2019.

[3] Y. K. Tun, C. W. Zaw, and C. S. Hong, "DownlinK power allocation in virtualized wireless networks," in *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, sep 2017.

[4] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[5] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, "Energy-efficient offloading for mobile edge computing in 5g heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[6] X. Chen, L. Jiao, W. Li, and X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, oct 2016.

비트코인 노드 메모리 풀 유사도 분석

고경찬 O, 이채현, 홍원기

포항공과대학교 컴퓨터공학과

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

An Analysis on Similarity of mempool on Bitcoin nodes

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

Department of Computer Science and Engineering, POSTECH

요 약

비트코인은 블록체인 기술을 기반으로 신뢰할 수 있는 공개 분산 원장을 구현한 암호화폐 플랫폼이다. 비트코인 네트워크에 참여하고 있는 노드들은 P2P 네트워크로 연결되어 있으며 새롭게 생성되는 트랜잭션들은 네트워크 전역에 배치되어 있는 노드들을 통해서 전달된다. 비트코인 노드들은 연결되어 있는 Peer 들에 따라서 다른 트랜잭션들을 수신하게 되는데 이는 각 노드들의 메모리 풀에 보관된다. 메모리 풀에 보관되는 트랜잭션들은 불법 거래 탐지 등의 연구를 위해서 필요한 실시간 트랜잭션이지만, 전파 지연으로 각 노드마다 수신하는 트랜잭션이 다르다는 문제가 있다. 본 연구에서는 비트코인 노드들이 메모리 풀에 저장하고 있는 트랜잭션들이 얼마만큼 차이가 있는지를 보이기 위한 실험의 결과를 보여준다.

I. 서 론

비트코인 [1]은 2008 년 사토시 나카모토라는 이름의 저자가 작성한 보고서를 통해서 처음으로 세상에 알려졌다. 비트코인은 블록체인기술을 기반으로 구현된 암호화폐 시스템이며 Gnutella, BitTorent 등과 같은 일종의 분산된 p2p(Peer to Peer) 환경에서 동작한다. 사토시 나카모토는 위의 보고서를 기반으로 비트코인 소프트웨어를 개발하였고 이를 2009 년 1 월에 오픈소스로 공개하였다. 그 이후로부터 비트코인 소프트웨어를 동작하는 노드(Node)들이 생겨나며 비트코인 네트워크가 형성되었다. 2019 년 4 월 현재 비트코인 네트워크에는 약 9500 개의 풀 노드(Full node)가 동작 중이다.

비트코인 네트워크에 참여 중인 노드들은 블록체인이라고 불리는 동일한 공개 거래(Transaction) 장부를 유지한다. 비트코인에서 트랜잭션들은 블록(Block)이라는 데이터 구조에 포함되고, 새롭게 생성된 블록은 이전 블록에 연결되어 블록체인을 연장시킨다. [3] 또한, 비트코인 노드들은 계속해서 새로운 트랜잭션을 생성할 수 있고, 생성된 트랜잭션은 네트워크를 통해서 다른 노드들에게 전파된다. 새로 생성된 트랜잭션을 수신한 모든 노드들은 각자의 메모리 풀에 이 트랜잭션을 임시로 보관한다. 이렇게 메모리 풀에 담긴 트랜잭션들은 다음에 생성될 블록에 포함되기를 기다리게 되며, 트랜잭션이 메모리 풀에 담겨있는 기간은 메모리 풀에 처음 들어온 시간부터 해당 트랜잭션이 블록에 포함될때까지 이다. 비트코인에서는 작업 증명(Proof-of-Work) 합의 알고리즘을 사용하며, 마이너(Miner)라는 특수한 노드가 수행하는 마이닝(Mining)을 통하여 새로운 블록이 평균적으로 10 분에 하나씩 생성되고 트랜잭션과 마찬가지로 네트워크를 통해서 전파된다.

비트코인의 초당 거래량은 약 7 TPS(Transaction Per Second) [4] 이지만 초당 생성되는 트랜잭션이 7 개라는 의미는 아니다. 비트코인 노드들은 초당 약 2~17 개의 새로운 트랜잭션을 수신하며 각 노드의 메모리 풀에 보관한다. [5] 비트코인 노드가 수신하는 새로운 트랜잭션의 수와 종류는 각 노드마다 다른데, 이는 P2P 네트워크 특성상 생성된 트랜잭션이 전파되어서 모든 노드들에게 전달되는 시간은 네트워크의 상태와 연결된 peer 노드들에 따라서 달라지기 때문이다. 메모리 풀에 담겨있는 트랜잭션들은 실시간 트랜잭션 분석, 불법 거래 탐지 등을 수행하기 위해서 중요한 역할을 차지하지만, 모든 노드들이 동일한 트랜잭션들을 가지고 있지 않다는 문제가 있다. 따라서 언급된 연구들을 수행하기에 앞서 본 연구에서는 비트코인 노드들의 메모리 풀을 모니터링하여 트랜잭션들을 수집하고 이를 이용하여 얼마만큼의 공통된 트랜잭션들을 가지고 있는지를 분석한다.

II. 관련 연구

Muhammad Anas Imtiaz [6]는 비트코인 네트워크에서 간헐적으로 발생하는 노드들의 churn 현상에 대한 논문을 작성했다. 해당 논문에서 저자는 비트코인 네트워크에서 발생하는 churn 의 특징을 분석

하여 Up / Down session length 에 가장 잘 맞는 분포를 지적했고, 실험을 통하여 churn 현상이 compact block protocol 의 성능을 상당히 감소시킨다는 것을 보였다. 또한, churn 현상 때문에 메모리 풀에 있는 트랜잭션들이 달라지고 이를 완화하기 위해서 메모리 풀의 효율적인 동기화 방법이 필요하다는 결과를 도출했다.

III. 메모리 풀 유사도 분석 실험

비트코인 노드들의 메모리 풀에 보관되어 있는 트랜잭션들 중에서 얼마만큼 동일한 트랜잭션이 있는지 알아보기 위한 실험을 진행했다. 실험 환경은 Dell PowerEdge R610 서버(Intel(R) Xeon(R) CPU X5650 @ 2.67GHz, 48G RAM) 4 대이다. 각 서버는 Bitcoin Core 클라이언트를 구동시켜 하나의 노드로서 비트코인 네트워크에 참여한다. 데이터 수집 방법은 RPC 를 이용해서 메모리 풀의 트랜잭션들을 모니터링을 수행하는 프로그램을 개발하고 이를 이용해서 4 개의 노드들로부터 각 메모리 풀에 저장되어 있는 트랜잭션들을 주기적으로 수집하는 것이다. 분석 방법은 수집된 트랜잭션 데이터들을 자카드 지수(Jaccard index) [7]를 이용해서 노드들의 메모리 풀에 들어있는 트랜잭션들의 유사도를 분석하는 것이다. 각 노드의 메모리 풀의 트랜잭션 수집은 10 분을 간격으로 10 번 시행되었고, 수집된 메모리 풀의 트랜잭션들을 집합으로 구성했다. 노드 1 의 메모리 풀에서 수집한 트랜잭션들의 집합을 Node1, 노드 2 의 메모리 풀에서 수집한 트랜잭션들의 집합을 Node2, 노드 3 의 메모리 풀에서 수집한 트랜잭션들의 집합을 Node3, 노드 4 의 메모리 풀에서 수집한 트랜잭션들의 집합을 Node4 라고 한다. 그림 1 는 총 10 번 수집한 데이터를 막대 그래프로 표현한 것이다. 그림 1 을 보면 동일한 시간에 수집한 메모리 풀의 트랜잭션 개수는 각 노드마다 차이가 있는 것을 확인할 수 있다. 비트코인에서는 노드에 연결된 peer 들이 해당 노드와 지역적으로 가까이 있음을 보장하지 않는다.

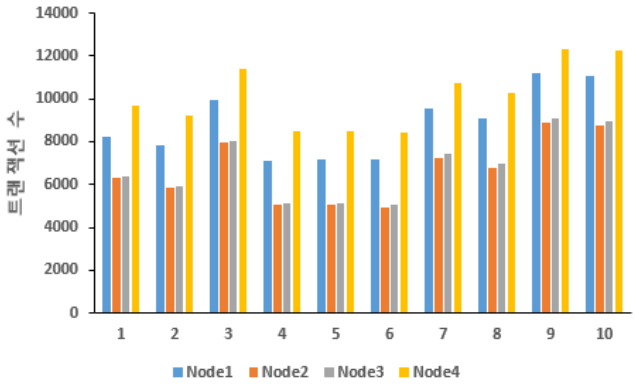


그림 1. 각 노드 별 메모리 풀 트랜잭션

따라서 연결된 peer 가 가까이에 있는 노드가 아닐 수 있으며 지리적으로 멀리 떨어진 노드일 수도 있는 것이다. 이러한 특징 때문에 그림 1 처럼 각 노드마다 메모리 풀에 있는 트랜잭션들에 차이가 발생한다. 또한, 각 노드들은 모든 시행에서 비슷한 트랜잭션 비중을 갖는 것을 확인할 수 있다. 그림 2 는 메모리 트랜잭션 유사도의 결과로서 각 시점에서 수집된 메모리 풀 트랜잭션 집합들을 합집합에 대한 교집합의 비율을 보여준다. 결과를 보면 노드들이 50~70%의 동일한 트랜잭션을 메모리 풀에 보유하는 것을 알 수 있다.

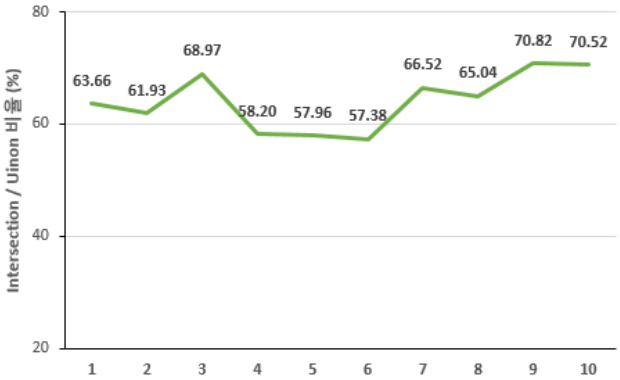


그림 2. 노드들의 메모리 풀 유사도

IV. 결론 및 향후 연구

비트코인은 P2P 네트워크로서 새로운 트랜잭션이 전달되기까지 노드마다 다른 전파 지연이 발생한다. 본 연구에서는 비트코인 노드의 메모리 풀에 저장되어 있는 트랜잭션들이 각 노드마다 차이가 있다는 것을 실험을 통해서 확인했다. 이를 통해서 각 노드들에 연결되어 있는 peer 들이 다르다는 것과 연결된 peer 들이 메모리 풀의 트랜잭션에 영향을 미친다는 것을 유추할 수 있었다. 향후 연구로는 비트코인 노드들이 동기화되는 시간을 비교하거나, 다른 암호화폐 플랫폼인 이더리움에서 노드들의 메모리 풀 트랜잭션들을 비교할 예정이다.

ACKNOWLEDGMENT

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학 ICT 연구센터지원사업의 연구결과로 수행되었음 (IITP-2019-2017-0-01633)

참 고 문 헌

[1] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008).
[2] “Bitnodes”, https://bitnodes.earn.com/, Accessed: April 24, 2019.
[3] Vallois, Valentin, and Fouad Amine Guenane. "Bitcoin transaction: From the creation to validation, a protocol overview." 2017 1st Cyber Security in Networking Conference (CSNet). IEEE, 2017.
[4] Müller, Paul, et al. "The Bitcoin Universe: An Architectural Overview of the Bitcoin Blockchain." 11. DFN-Forum Kommunikationstechnologien. Gesellschaft

für Informatik eV, 2018.
[5] “Transaction Rate”, https://www.blockchain.com/charts/transactions-per-second?timespan=30days, Accessed: April 24, 2019.
[6] Imtiaz, Muhammad Anas, et al. “Churn in the Bitcoin Network: Characterization and Impact.”
[7] “Jaccard index”, https://en.wikipedia.org/wiki/Jaccard_index

강화학습을 이용한 UAV-EDGE 협업 태스크 오프로딩 방안 연구

김기태, 홍충선*
경희대학교 컴퓨터공학과
glideslope@khu.ac.kr, *cshong@khu.ac.kr

A Study on UAV-Edge Joint Task offloading Scheme using Reinforcement Learning

Kitae Kim, Choong Seon Hong*
Kyung Hee University

요 약

UAV(Unmanned Aerial Vehicle)는 이동이 가능하며 카메라와 센서, 컴퓨팅자원, 통신기기를 탑재할 수 있다는 장점을 가지고 있다. 따라서 5세대 통신에서 UAV가 중요한 역할을 할 것이라고 예측되고 있다. 이러한 통신기술과 UAV의 장점을 결합하여 재난현장에서의 신속한 대처가 가능하며 공연장과 스포츠 경기장과 같은 트래픽 수요가 많은 곳에서 이동 기지국으로 활용 될 수 있다. 또한 탑재된 카메라나 센서를 활용해 다양한 데이터 수집 및 수집 된 데이터 기반 서비스 제공이 가능하다. 하지만 이러한 서비스는 빅 데이터 처리와 머신러닝과 같은 고성능의 컴퓨팅 능력이 필요하며 컴퓨팅 자원의 한계가 있는 UAV에서 이러한 서비스를 홀로 처리하기에는 무리가 있다. 따라서 본 논문에서는 태스크 발생 시 다수의 엣지 서버와 UAV의 현재 상태를 고려하여 태스크 수행을 위한 엣지-UAV 매칭 및 최적의 UAV의 위치선정에 대해 제안한다.

Ⅰ. 서 론

고 대역폭, 저 지연 서비스를 제공할 수 있는 5세대 통신의 등장과 다양한 스마트 디바이스, IoT 센서, 드론과 같은 다양한 기기의 등장으로 수 많은 데이터와 이를 이용한 다양한 어플리케이션이 등장하였다. 이러한 어플리케이션들은 가상현실과 증강현실, 머신러닝, 빅 데이터처리와 같은 고성능 컴퓨팅 자원이 필요하며 스마트폰이나 무인항공기와 같은 기기에서 실행되기에는 무리가 있다. 이러한 문제를 해결하기 위하여 기지국이나 AP와 같은 네트워크의 가장자리(Edge)에 컴퓨팅 자원을 위치시켜 모바일 유저나 IoT 센서들이 인접한 노드의 컴퓨팅 자원을 활용할 수 있는 MEC(Mobile Edge Computing) 기술이 등장하였다[1].

이동 가능한 무인 항공기는 카메라 뿐 만 아니라 다양한 센서와 통신 기기를 탑재할 수 있다는 점에서 큰 주목을 받고 있다. 이러한 무인 항공기는 앞서 언급하였던 엣지 노드가 될 수 있으며 무인 항공기를 이용해 재난 현장에서의 신속한 대처와 통신 수요량이 많은 지역에서 이동식 기지국으로서 활용이 가능하다[2]. 특히 이동하면서 다양한 사진이나 영상과 같은 다양한 데이터 수집을 및 분석을 통하여 이상 상황 탐지, 농작물 성장추이, 기상 예측 등과 같은 데이터 기반 서비스 제공이 가능하다. 하지만 앞서 언급하였듯이 이러한 데이터 분석은 빅 데이터 처리 기술이나 머신러닝과 같은 고 성능 컴퓨팅 자원을 요구하며 자원의 한계가 있는 UAV에서 이러한 태스크를 수행하기에는 무리가 있다. 따라서 본 논문에서는 태스크 발생 시 Task와 UAV의 위치, 상태에 따른 Task-UAV 매칭과 엣지 노드와 태스크 수행 협업을 위한 최적의 엣지 노드 선택과 이를 위한 UAV의 최적의 위치 선정에 대한 연구를 진행하였다.

Ⅱ. 시스템 모델

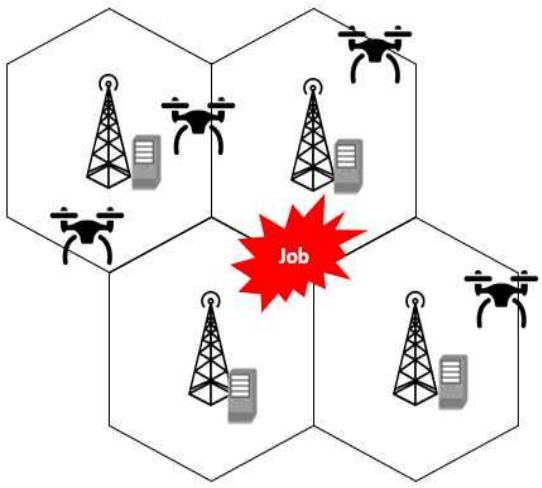


그림 1 . 시스템 모델

본 논문에서는 그림1과 같이 4개의 엣지 노드와 무작위의 위치에 배치된 4대의 UAV를 고려한다. 여기서 모든 UAV

는 고도는 일정한 것으로 가정한다. Task 또한 무작위의 위치에서 발생한다. UAV의 상태로는 각각의 현재의 위치 값과 배터리 상태를 고려한다. 엣지 서버에서는 현재의 유휴 자원상태를 고려한다. Task는 Task의 크기와 Task를 실행하는데 필요한 CPU Cycle, 그리고 Task의 타입을 고려한다. 태스크는 2가지의 타입으로 구분되며 Data Rate와 실행 시간을 보장이 필요한 Task(1)와 그렇지 않은 Task(0)로 나뉜다.

종류	변수	설명
UAV	B_{UAV}	현재 배터리(J)
	L_{UAV}	UAV 현재 위치
Task	T_{cpu}	수행에 필요한 CPU Cycle
	T_{size}	Task Input 사이즈
	T_{type}	Task 타입(0,1)
	$T_{location}$	Task 발생위치
Edge	E_{cpu}	CPU 사용률

표 1 . UAV-TASK-EDGE에서 고려되는 변수

UAV 에너지 소비 모델

랜덤한 위치에서 Task가 발생하면 UAV는 Task의 위치에 가서 Task 수행을 위한 데이터 수집 후 함께 협업 할 엣지 서버의 위치까지 비행해야 한다. 이 때 소비되는 UAV의 에너지 모델[3]은 아래와 같다.

$$E = \frac{p^{\min} d}{v\eta}$$

수식 1. 이동 거리에 따른 UAV 에너지 소비 모델

수식1.에서 v와 d는 각각 UAV의 속도와 총 비행 거리를 나타내며 η은 Power Efficiency를 나타낸다. $p^{\min}$은 UAV가 움직이기 위한 최소한의 에너지를 나타내며 아래 수식2와 같이 정의된다.

$$p_{\min} = (v_1 + v\sin\beta)T$$

수식 2. UAV 추진을 위한 최소 에너지

여기서 T는 추진하기 위한 힘을 나타내며 수식3과 같이 정의된다.

$$T = mg + f_d$$

수식 3. UAV의 추진력

m은 UAV의 무게 g는 중력가속도 f_d는 공기에 의한 항력을 나타낸다. 수식2에서 v_1은 추진력 T를 속도로 수식 4로 정의 된다.

$$v_1 = \frac{2T}{qr^2\rho\sqrt{(v\cos\beta)^2 + (v\sin\beta + v_1)^2}}$$

수식 4. 추진력 T를 위한 속도

통신 및 컴퓨팅 모델

최적의 위치에서 UAV로부터 엣지 노드에게 태스크 오프로딩을 위하여 UAV-Edge 간 Uplink Data Rate를 측정한다. 본 논문에서의 Data Rate[4]는 아래와 같이 측정된다.

$$DataRate = B\log_2(1 + \frac{pc}{\sigma^2 + I})$$

수식 5. UAV-Edge간 Data Rate

수식2.에서 B는 대역폭, p는 transmission power, c는 channel gain, σ^2는 noise power를 나타낸다. 수식2의 Data Rate를 이용해 엣지로의 오프로딩 시 컴퓨팅 소요시간은 아래의 수식6과 같이 모델링 될 수 있다.

$$T = \frac{T_{cpu}}{A_{cpu}} + \frac{T_{size}}{DataRate}$$

수식6. 총 컴퓨팅 소요시간

총 컴퓨팅 소요시간은 Data Rate에 따른 엣지에서의 태스크 인풋에 대한 전송시간과 태스크 수행을 위해 엣지에서의 Task 수행시간의 합으로 정의 된다. 따라서 A_{cpu}는 엣지에서 태스크 수행을 위해 할당된 CPU Cycle를 의미한다.

Ⅲ. 제안사항

본 논문에서는 Q-Learning을 이용하여 Task의 요구에 맞게 UAV의 에너지 소비량과 Uplink Data Rate에게 적절한 가중치를 주어 적절한 위치에서의 Task의 요구를 만족시킬 수 있는 오프로딩 기법을 제안한다.

Deep Q-Learning

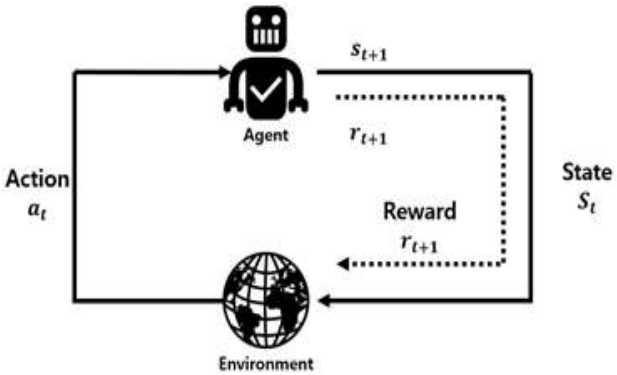


그림 2. Q-Learning

강화학습의 한 종류인 Deep Q-Learning[5]은 학습 에이전트(Agent)가 상태를 관찰 후 어떠한 행동(Action)을 취함

으로써 그에 따른 보상(Reward)를 받는다. 이 과정을 통하여 에이전트는 미래의 보상 값을 극대화하기 위한 의사결정을 학습한다. 따라서 Q-Learning에는 보상을 어떻게 주느냐가 가장 중요하다. 이러한 Q-Learning은 2015년 등장했던 인공지능 바둑 기사인 알파고에 적용된 학습 알고리즘이기도 하다. 본 논문에서 Q-Learning을 위한 상태는 앞서 표 1에서 소개한 모든 UAV의 상태와 위치, Task의 위치 및 특성, 협업 가능한 모든 Edge Server의 상태를 통해 계산된 에너지 소비량과 Data Rate, 오프로딩 시간을 통해 정의된다. 이러한 상태를 관찰한 에이전트는 어떠한 UAV가 Task를 수행하여 어떤 엣지와 협업하는지 결정하는 행동(Action)을 취하여 행동에 상응하는 보상함수를 받게 된다. 적절한 보상을 위하여 본 논문에서의 보상함수는 앞서 Ⅱ에서 실제 Task의 요구사항을 만족시키는데 영향을 미치는 UAV에너지 소비 모델과 통신·컴퓨팅 모델에 따라서 보상함수가 계산된다. Task의 2가지 특성에 따라서 만족시켜야 하는 요구사항이 다르므로 Task의 특성에 따라 에너지 소비 모델과 통신·컴퓨팅 모델에 가중치 α,γ,δ 를 정하여 보상함수 계산을 한다. 가중치는 Task의 종류에 따라 실행시간이 보장되어야 하는 서비스의 가중치를 높이고 다른 가중치에 대해서는 낮은 가중치를 주는 방식으로 적용한다. 공통적으로 보상함수는 UAV 이동을 위한 에너지 소비를 줄이며 통신효율이 좋은 위치를 찾아내는 것을 목표로 한다.

$$Reward = \alpha\frac{1}{E}+\gamma\frac{DataRate}{\delta T}, \quad \alpha+\gamma+\delta=1$$

수식 7. 보상함수

이러한 보상함수를 통해 학습이 진행된 에이전트는 Task가 발생함에 따라 앞서 언급한 상태를 Input으로 하여 어떠한 드론이 Task를 받아 어떠한 엣지 노드와 협업을 했을 때 가장 보상 값이 클지 계산하는 방법으로 최적의 정책을 학습한다. 알고리즘의 Output으로는 Task를 수행할 드론과 협업할 Edge 노드 그리고 Edge 노드와 협업하기 위하여 이동해야할 위치가 나온다.

Ⅳ.시뮬레이션

변수	값
ρ	$1.225kg/m^3$〔6〕
f_d	9.6998N〔6〕
q	4〔6〕
r	0.254m〔6〕
η	70%〔6〕
v	1.49m/s〔6〕
B	〔1,10〕MHz
p	〔0.5,1〕W
σ^2	$2\times10^{-13}W$
c	$127+30\times\log d$

표 2. 시뮬레이션 변수

표2는 Ⅱ절의 수식들을 계산하기 위한 변수를 나타낸다. 위치별 상이한 통신 상태를 위하여 Data Rate 계산을 위한 변수들은 일정한 범위 내에서 임의의 값을 가지게 된다. 시뮬레이션을 위해 파이썬과 강화학습을 위하여 Tensorflow를 이용하였다. 시뮬레이션을 위하여 200X200의 2차원 평면을 만들어 4개의 엣지 서버와 4대의 임의의 위치의 드론, 1개의 임의의 Task를 발생시켰다.

시뮬레이션 결과

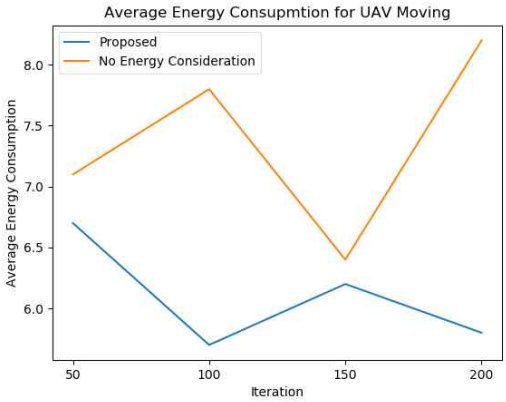


그림 3. UAV의 평균 이동 에너지 소비량

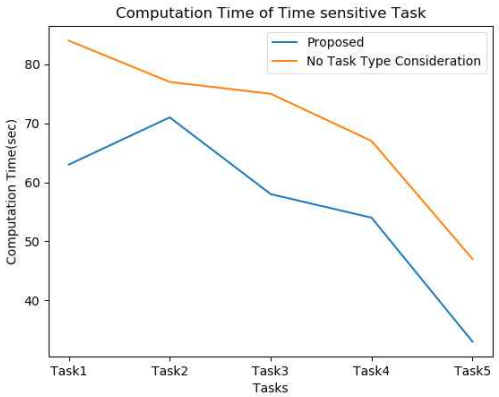


그림 4. Task에 따른 컴퓨팅 타임

그림3은 본 논문에서 제시한 알고리즘에 대해 UAV의 에너지 소비량을 고려하지 않았을 때와 고려했을 때 평균 에너지 소비량을 비교하였다. Task의 컴퓨팅 요구를 맞추는 것도 중요하지만 UAV의 배터리도 한계가 있기 때문에 두 가지를 동시에 고려하는 것이 중요하다. 에너지 소비량을 고려하였을 때 UAV가 Task로의 이동 이후에 엣지까지 이동하는 평균 에너지가 적음을 알 수 있다. 그림4는 동일 Task 5개에 대하여 실행시간의 보장이 필요한 Task와 그렇지 않은 태스크를 고려하지 않았을 때와 고려하였을 때의 평균 컴퓨팅 시간을 비교하였다. Task 타입을 구별하였을 때 큰 가중치가 적용이 되어 컴퓨팅 자원이 풍부하고 적절한 위치에서의 오프로딩이 이뤄지기 때문에 컴퓨팅 시간이 더 적게 걸리는 것을 확인 할 수 있었다.

Ⅴ. 결론

본 논문에서는 Task 발생 시 UAV-Edge간의 협업을 위한 강화학습 기반의 Task 오프로딩 기법을 제안하였다. 드론이 어느 정도의 컴퓨팅 자원을 제공 할 수 있지만 컴퓨팅 자원의 제약과 배터리 문제로 드론 또한 엣지의 컴퓨팅 자원과의 협업이 필요하다. 이 때 드론의 에너지 문제와 컴퓨팅을 가장 잘 수행할 수 있는 주변의 엣지 서버를 찾아 가장 효율적인 위치에서 오프로딩을 하기 위하여 강화학습을 적용하였다. 향후에는 다수의 Task가 동시에 일어났을 때의 처리와 UAV-UAV간 협업, 다수의 엣지 서버와의 협업을 고려하여 연구를 진행 할 계획이다.

ACKNOWLEDGMENT

이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2019-0-01287, 분산 엣지를 위한 진화형 딥러닝 모델생성 플랫폼). 또한 이 논문은 2019년도 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] 박준민." Cloud to Rain, 엣지 컴퓨팅이 가져올 변화," 정보통신산업진흥원, pp. 4-5, 2018

[2] Sarder Fakhrul Abedin, Md. Golam Rabiul Alam, Choong Seon Hong, “Learning Based Intelligent IoT Task Offloading and Resource Allocation in UAV-assisted Fog Network”, 2017년 한국컴퓨터종합학술대회 논문집

[3] Momena Monwar, Omid Semiari, Walid Saad "Optimized Path Planning for Inspection by Unmanned Aerial Vehicles Swarm with Energy Constraints“, arXiv1808.06018, Aug, 2018

[4]Min Chen, Yixue Hao, “Task Offloading for Mobile Edge Computing in Software Defined Ultra-Dense Network”, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL36, NO3, March 2018

[5] “Q-Learning”,https://en.wikipedia.org/wiki/Q-learning

[6] J. K. Stolaroff, C. Samaras, E. R. ONeill, A. Lubers, A. S. Mitchell, and D. Ceperley, “Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery” Nature Communications vol. 9, no. 409, pp. 1 – 13, February 2018.

Caching in Named Data Networking with an Open Source Edge Computing Framework

Muhammad Atif Ur Rehman[1], Rehmat Ullah[2], and Byung-Seo Kim[3]

Dept. of Electronics & Computer Engineering, Hongik University[1,2]

Dept. of Software & Communication Engineering, Hongik University[3]

atif_r@outlook.com[1], rehmat_ciit@hotmail.com[2], jsnbs@hongik.ac.kr[3]

Summary

The two emerging technologies Named Data Networking (NDN) and Edge Computing (EC) are considered as the most representative technologies for the future Internet. NDN directly forwards an application layer name on network layer. Thus, it mitigates the requirements of an individual and complex network addressing schemes for devices. Moreover, in NDN, the intermediate nodes and/or routers cache the content inside their memory, so that the request for same content could be responded quickly in future. EC, on the other hand, offers computation, data, and application services in close proximity of end users which results in lower delay and high speed of task execution. In this paper, therefore, we integrate NDN with EC in order to evaluate the impact of NDN caching with EC. For the evaluation, we employ ndnSIM simulator along with our self-developed .NET based open source EC framework. The measurements show that enabling NDN caching with EC reduces the traffic on Edge node.

1. Introduction

Named-data networking (NDN) [1] is an enhanced version of the Content Centric Networking (CCN) architecture. Similar to CCN, NDN also follows the interest/data packet exchange to obtain any particular data. Each node in NDN maintains three types of data structures: 1) a Content Store (CS), 2) a Pending Interest Table (PIT) and 3) a Forwarding Information Base (FIB). NDN adopts pull-based consumer driven communication model where the consumer(s) request for the content which is forwarded by the content router (CR) if not cached in the CS of the CR. The Interest packet reaches to the content provider which then respond with Data packet the way back to the consumer following the breadcrumb path that created with the help of PIT. NDN has an adaptive forwarding mechanism and employs FIB that keeps information for routing and forwarding planes.

Edge computing (EC) [2] refers to the enabling technologies allowing computation to be performed at the edge of the network. Here the "Edge" means any computing and network resources along the path between data sources and cloud data centers. The core idea of EC is to bring the resources at the network edge such as computation and storage. The storage resources are moved closer to the IoT/end user devices to reduce the data traffic and the response latency, and to facilitate the resource intensive IoT applications. EC has relatively smaller computation capacity compared to cloud computing, however, takes advantage of short access distance, flexible geographical distribution, and relatively richer computational resources than end user mobile /IoT device(s).

Nodes in NDN cache the data and even results of functions/services and make them available to other consumers without performing the computation again and again [3]. NDN over EC may achieve latency requirements by providing data and services closer to end users via caching the content and computed results (from edge node) as well. Therefore, NDN not only provides content caching but also functions/code caching in order to avoid re-request the content or re-execute the function/code.

To evaluate the behavior of NDN caching with EC, we use ndnSIM simulator for the NDN network and self-developed .NET based open source EC framework for EC application. Further detail about EC framework is presented in the next Section.

The rest of this paper is organized as follows. Section 2 briefly explains the architecture of EC application. The experimental setup is presented in Section 3. In Section 4, we present the results and discussion and finally, we conclude this paper in Section 5.

2. Edge Computing Application Architecture

EC application comprises of seven different layers as illustrated in Figure 1. The API layer (Layer 1) is responsible to satisfy the requests from NDN network (ndnSIM). The web layer (Layer 2) shows a graphical user interface for real time data display. Service Layer (Layer 3) coordinates data and services between an API layer and the Data Access Layer (DAL) (Layer 4). It contains all the business logics of the services on which the decision might be taken. DAL is responsible for the data management, typically using a database such as SQL, MySQL, Oracle, MongoDB, etc. In our implementation, we have used SQL database. The unit test layer (Layer 5) is provided in order to ease the testing of an application. In

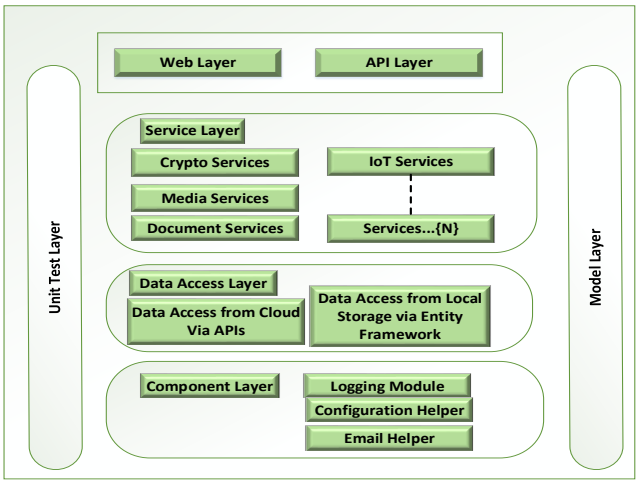


Figure 1: EC application architecture

component layer (Layer 6) we have provided those code modules that are shared among different layers. Model layer (Layer 7) comprises of various type of classes used for carrying data among different layers.

We open source all the source code [4] to the research community for reuse and further improvement.

3. Experimental setup

In our experimental setup, the edge application is hosted on Edge node with specification of 16 GB RAM, core-i7, 4710HQ-CPU and 2.40 Ghz core. For NDN network, we generate requests from ndnSIM to the edge node. In order to do that, we have modified some of the code in *ndn-producer.cpp* file and *ndn-producer.hpp* file. Our custom function which generates the request is using *boost/asio.hpp* library and its *class asio::ip::tcp*. ndnSIM is running on Linux using VMware. The VMware has 8 GB RAM and 4 core CPU.

4. Results and Discussion

EC with NDN is a promising approach to increase the performance of several applications by employing in-network caching. In NDN, each intermediate node cache content or computed results (from edge node) inside their memory for fast access.

To evaluate the behavior of caching, the requests that are generating in ndnSIM [5] are following Zipf-Mandelbrot distribution with following parameters values: alpha=1.12 and q=5. The content catalogue size is varied from 1000 to 5000 with an interval of 500. The interest frequency is set at 100 interest per second and simulation time is 120 s. Figure 2 depicts the behavior of caching with various content catalogue size. When content catalogue size is set at 1500, the cache hit ratio is 22.77% which means 2733 number of requests out of 12000 satisfied from local cache nodes inside NDN network. Thus, 22.77% of traffic is reduced at the edge node using in-network caching feature of NDN. Moreover, when content catalogue size is set at 5000, the hit ratio is 13.65% which

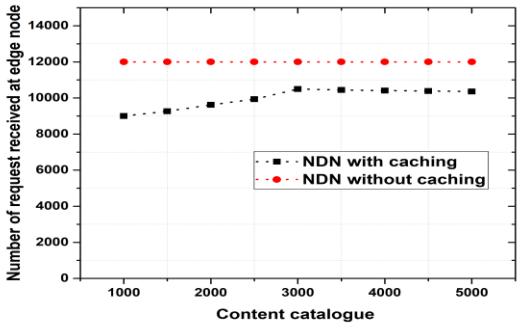


Figure 2: Number of requests received at edge node as function of content catalogue size

means 1638 number of requests out of 12000 satisfied from local cache nodes inside NDN network. Thus, 13.65% of traffic is accommodated from the nodes in the NDN network. From this experiment, we have observed that in case of IoT, NDN is a promising approach with EC to reduce the traffic on the edge node.

5. Conclusion

In this paper, we have evaluated the behavior of NDN caching by employing our EC application for edge node and ndnSIM for the NDN network. The experimental results show that NDN with EC reduces the traffic on the edge node.

ACKNOWLEDGMENT

This research was supported in part by the National Research Foundation of Korea (NRF) through the Korea Government (2018R1A2B6002399) and in part by the International Research & Development Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT. (No. NRF-2018K1A3A1A39086819).

References

[1] Jacobson, V., Smetters, D., et al. “Networking named content”, *Proceedings of the 5th international conference on Emerging networking experiments and technologies. ACM* 2009 pp. 1-12.

[2] R. Ullah, S. H. Ahmed and B. Kim, “Information-Centric Networking with Edge Computing for IoT: 685 Research Challenges and Future Directions,” in *IEEE Access*, vol. 6, pp. 73465-73488, 2018.

[3] Manolis Sifalakis, et al. “ An information centric network for computing the distribution of computations” *In Proceedings of the 1st (ACM-ICN ’14),* ACM, New York, USA,2014, pp. 137-146.

[4] R. Ullah, M. A. U. Rehman and B. Kim, “Design and Implementation of an Open Source Framework and Prototype for Named Data Networking-Based Edge Cloud Computing System,” in *IEEE Access*, vol. 7, pp. 790 57741-57759, 2019.

[5] Spyridon Mastorakis, Alexander Afanasyev, and Lixia Zhang. On the evolution of ndnSIM: “An open-source simulator for NDN experimentation”. *ACM SIGCOMM Computer Communication Review* 47, 3 (2017), 19–33.

우주 기상 관측 데이터를 위한 분산 저장소

요가안드리안*, 주홍택*
*계명대학교 컴퓨터공학과

yoga.andrian@lapan.go.id, juht@kmu.ac.kr

Decentralized Storage System for Space Weather Observation Data

Yoga Andrian*, Hongtaek Ju*
*Dept of Computer Engineering, Keimyung University.

Abstract

Indonesia is one of the countries in ASEAN who is actively researching aeronautics development and space weather activities. Represented by National Institute of Aeronautics and Space (LAPAN; https://lapan.go.id), we develop a system that provides actual information and prediction related to space weather called Space Weather Information and Forecast Services (SWIFtS; http://swifts.sains.lapan.go.id). SWIFtS have supported by the data storage system that serves data near real-time. Since the data served by centralize model which collected on one single server, problems emerge when the researchers need it for data processing and making a forecast to update the content in SWIFtS website. The system incapable of providing the latest data due to server down. Therefore, we propose a new system that utilizes the decentralized model for storing data using the Inter Planetary File System (IPFS). Our scheme will increase the data availability by spreading it into nodes through a peer-to-peer connection. Other unused resources would be useful and no single point of failure. This system utilizes the Inter Planetary Linked Data (IPLD) for the data structure, Distribution Hash Table (DHT) to coordinate and maintain metadata, and BitSwap protocol for exchanging data between peers. We are also applying IPFS-Cluster tool to automatically distribute the data to other peers.

Ⅰ. Introduction

As the advancing of technology, many countries have developed research related to space weather. Indonesia is one of the countries in ASEAN that take a contribution to it. At present, Indonesia, represented by the National Institute of Aeronautics and Space (LAPAN; https://lapan.go.id), is actively developing a system that providing information and forecast related to space weather activities called Space Weather Information and Forecast Services (SWIFtS; http://swifts.sains.lapan.go.id). This system is specifically intended for users who use application that utilizes satellite service or radio wave technology such as for navigation, radio communication and so on. SWIFtS is expected to consistently present the actual information for user optimally. For this reason, the data availability from space weather observation instruments is needed to maintain information renewal. SWIFtS is supported by near real-time data storage systems originating from various instruments located in several observation stations in Indonesia. The daily information provided by SWIFtS demands the data storage system be continuously running well.

The current system uses a centralized model, each time PC collected data from the instrument, it sent to a server located in observation stations. In fig 1, we can see there are 8 observation stations (next we call it as a site) spread in Indonesia and 1 central datacenter. Researchers in Bandung, which datacenter located, need actual data for data processing and making forecasts to update on SWIFtS website. Problems arise when servers in Bandung must provide the latest data while the site server is down and vice versa. The researchers getting trouble to conduct their research as well as the renewal of information on SWIFtS website is hampered. The factors that cause server down as such due to power outages, loss of network connection or OS failure.

The current system employs crontab for running a script program in the interval of every 15 minutes. Bandung server will sync all data from all site server in one time using default SSH port by Rsync tool. Thus, this process makes queue jobs that have to be done while the variance of data from each site and instrument should be stored in real-time and safe without damage. In addition, of course, it lowers throughput in sending data because of runs at the same time on one port and low throughput affects the duration of time in data synchronization due to the high network load, so that takes longer. Delay in updating files also occurs when files with small size

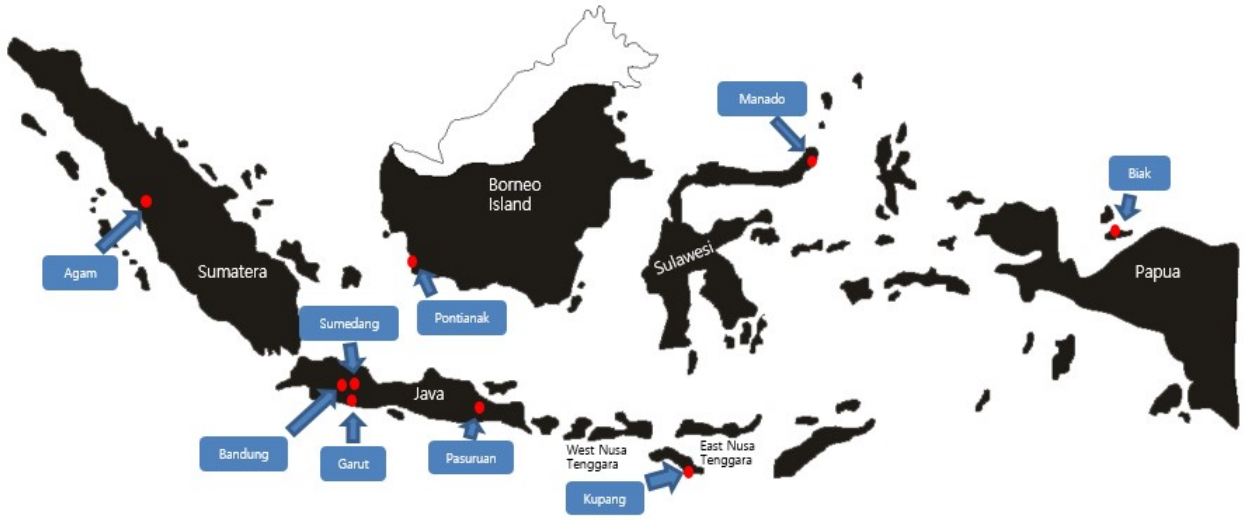


Fig 1. Observation Stations Map

cannot be sent immediately because waiting for the larger one queue has finished.

To address those problems, we need to propose a new system that can increase the percentage of data availability in real-time. The decentralized storage system is a method of storing data by encrypting and distributing data across a decentralized network. It does not rely on central services provider for data store [1]. Decentralized storage may increase the percentage of data availability by spreading files to be stored on each host that is connected by peer-to-peer. So, this method would reduce user dependency on a single server where each host can exchange files directly as well as have roles as a client and server. In addition, encryption may become an additional advantage for compressing files and protecting from data corruption. Thus, data generated by the instrument of sites will be kept on each data collection server continuously without concerning about single point failure and also providing better security of data[2].

The Inter Planetary File System (IPFS) is a peer-to-peer decentralized system protocol for distributing files through connected computing devices in the same system of files. IPFS serves a high-throughput data with content-addressed block storage model, no central server, as well as the data is distributed and stored in spread location [3][8]. In this paper, we apply IPFS as a decentralized system service for storing space weather observation data with a detailed explanation of how works of it. We also add directory watcher for the base trigger of our system in order to make data update process real-time continuously run.

Ⅱ. Related Works

This section reviews related works which leverage decentralized network approaches to store the data. But they need user as an actor to upload or download the data manually. In addition, most of the papers explained about utilization of IPFS as their storage system combines with blockchain. In addition, they didn't explain detail information on how IPFS works.

Sia is one of simple decentralized storage system proposed by Vorick, et.al [10]. They need clients as a user to upload and download files in Sia network, so there is no automation process in getting or putting files. Besides, there is Storj, a peer-to-peer cloud storage network [11]. Wilkinson assisted by some contributors has built Storj by proposing a Proof-of-Storage. But this system also needs actors to upload and download files manually from the system. Similar to Sia and Storj, Maidsafe comes with other demand for storing data in the decentralized and secure network [12] and need manually process to store and retrieve the data. Sia, Storj and Maidsafe rely on blockchain network and more commercial focus, need cryptocurrency to use their service. In 2018, Nygaard [13] leveraged IPFS-Cluster to limit the replication of data only for peer members. His proposed system also needs the client as actors for data storing manually. However, our proposed system does not need cryptocurrency or token since no commercial oriented in this system and we focus on file exchange without coin. Moreover, this system also no manually process to store the data due to various data produced by the instruments in kinds of range time continuously such as every 5 minutes, 15 minutes, hourly and daily. It would be impossible for conducting by human as manually process.

Built a scale-out Network Attached Storage (NAS) and IPFS to store an object spread through Fog/Edge infrastructure has been proposed by Confais, et.al [9]. This system implemented a Proof-of-Concept using RozoFS with perform some modifications in the IPFS source code. But, this system doesn't explain the detail explanation about how IPFS works and what protocol that IPFS uses. Another work proposed decentralization for big data by Brisbane [4]. He leverages IPFS changing Hadoop Distributed File System (HDFS) as the filesystem. But, he didn't give detail explanation about how IPFS store and retrieve

Table 1. Observation Stations Status in Q1 of 2019

No	Observation Stations	Total Amount of Daily Data (MB)	Bandwidth (Mbps)	Internet Connection Link Type	Frequent of Server Down (Jan – Apr 2019)		
					Network Offline (FO Cut)	Power Outages	OS Failures
1	Garut, (Java Island)	21	5	Fiber Optic	0	0	0
2	Pasuruan (Java Island)	39	7	Fiber Optic	4	0	0
3	Biak (Papua Island)	46	5	Wireless	1	30	0
4	Agam (Sumatera Island)	76	4	Wireless	2	0	0
5	Kupang (Nusa Tenggara Island)	106	5	Wireless	No data	No data	No data
6	Pontianak (Borneo Island)	110	7	Fiber Optic	3	4	0
7	Manado (Sulawesi Island)	151	1	Fiber Optic	No data	No data	No data
8	Sumedang (Java Island)	175	4	Fiber Optic	4	0	0

the data. In our paper, we give the detail explanation of how does IPFS work as a distributed filesystem using a P2P network.

Ⅲ. System Design

Decentralized system changes the paradigm that relies on a server as the highest level on a centralized system hierarchy. A host function only as clients initially, but its status raised equal to the server's role. So in a decentralized system topology, all nodes can act as clients and servers. According to this, IPFS has no single point of failure since nodes do not need to trust each other. IPFS allows any node who has connected in the network to be able to retrieve and store data for themselves by "pinning" content actively [4].

We utilize IPFS as our core system to store data and distribute it to other nodes. We include the instrument PC as a part of our system because it is the initial contributor that collect data from the instrument. They have two types of Operating System installed, Windows and Linux. Figure 2 shows the architecture of proposed system. The data communication module between server and PC are using TCP/IP. For data transfer module, we divide based on PC's operating system. FTP used for data transfer to the Windows whereas SFTP for the PC which installed Unix OS. Each of them has a data collector as a place where data stored. On the IPFS node, we employ two services that essential to keep the latest data is available. In figure 2, the scheduler and synchronization service is for data mirroring and another one is data collector watcher service for uploading data to the IPFS network. The server has to run IPFS binary to manage its resource as an IPFS node. LibP2P is the major part of our system that has some charge such as allow for data and communication transports, create a distributed hash table and file exchange in the system. For data communication module, link between PC and server connected by TCP/IP.

Our system uses Virtual Private Network (VPN) to build link communication in our underlying network. VPN is useful to keep the connection securely by creating a secure virtual tunnel through the internet. In our system, we only use the VPN connection provided by the vendor. IPFS is applied to all servers (sites and central) and use IPFS-Cluster to connect them as members in one cluster. Each server has a unique identifier called node ID and using multiaddr formatted byte string to communicate among nodes in the overlay network. Multiaddr is used to support addresses for any network protocol and also support for encapsulation address [3]. Besides, IPFS-Cluster is used to coordinate between IPFS daemons running on different hosts [5] and give the limitation of file distribution only flow in our environment.

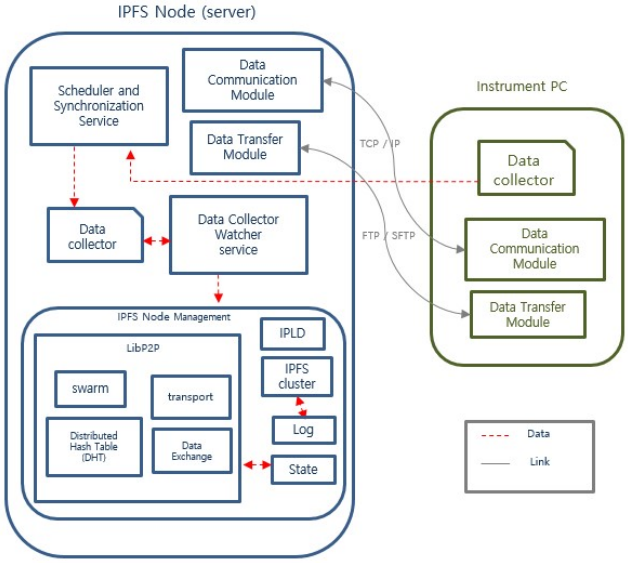


Fig 2. System Architecture

Since each site has various instruments that produced data with multiple ranges of time, we make them as the normal peers and the central server as a peer leader in the cluster consensus. When each node needs for exchanging files, a peer-to-peer connection is built between them, so a node can connect to another directly (see fig. 3). For example, the node in Biak, where different island with Bandung, has collected the latest file from instrument CADI, it's ready for distributing to other nodes, i.e. node Bandung. As they connected in a cluster of peers, peer Bandung request for that file from Biak, and P2P connection is built between them, peer Bandung can

get the file directly from peer Biak. This occurs for all peers too, not only Bandung and Biak. All peers can connect each other directly when a peer asks another peer to serve the requested file. Bandung is important as a main of our datacenter and has the major users who need the data, so Bandung has to collect all files by replicate it each time new file created. But, we won't replicate file to all peer in consideration of network and storage cost. Each node has distinct characteristics in bandwidth capacity, link connection, and the total amount of data daily (see table 1). We only replicate the data to other three nodes for our default system similar to replication factor in Hadoop Distributed File System (HDFS) [15].

According to table 1, we consider choosing other nodes for storing the data based on the same geographical location with Bandung in Java Island. This will impact in reducing the time when Bandung needs to get the file from other nodes which offline status. Moreover, we consider selecting the other nodes which have the lowest total size in collecting data for daily, nodes with higher bandwidth capacity than others and prefer to choose nodes that connected by fiber optic link. So, based on these considerations, we select the node in Garut and Pasuruan as the second and third center of data replication. Currently, node Garut has 5Mbps for the bandwidth capacity with fiber optic connection and collecting 21MB data in daily. Meanwhile, node Pasuruan has 7Mbps for the bandwidth capacity by fiber optic link and collecting 39 MB data in daily.

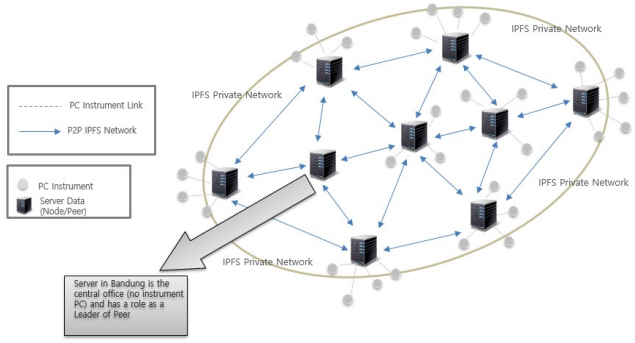


Fig 3. Network Diagram

Fig 4 depicts how the data stored in the system, start from the instrument PC that collects the data from the instrument. In our system, we use two methods for collecting data between machines. First, when a new file created by the instrument and stored in PC, the site server will run scheduler to synchronize it from PC every 6 minutes. Secure Shell (SSH) port is used for the PC which installed Linux OS to sync the data by Rsync and FTP for the PC which has Windows OS. Second, we use filesystem watcher to observe the root directory of data in the site server. So, this watcher will force the server to upload the latest file into IPFS network each time it is created. Afterward, the server as an IPFS node, will store the file as an object in the local repository and put the metadata of object in the DHT.

IV. File Distribution

Basically, we develop a system from the existing and focus only for changing the storage system model by implement IPFS to increase data availability and provide real-time data for supporting SWIFtS system. In order to keep the data availability, data replication must be prepared. Moreover, replicated files also can bring positive impact in our network workload where it splits over multiple source. In our cluster, we replicate data only for three numbers of copies to other peers, Bandung is selected because of a central data center with the highest resource, Garut and Pasuruan are chosen due to some reasons such as geographical distribution, amount of data collected, network and storage cost.

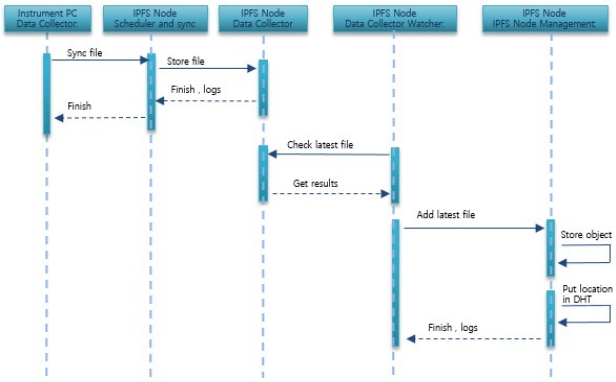


Fig 4. Sequence Diagram of Store the Files

So, when a node is not able to provide the block, it still provided by others. In IPFS configuration, we can set the upper limit for the size of IPFS repository that stored in our local disk called storageMax. IPFS-Cluster makes a priority for block replication based on the high free space of peer. Therefore, we consider setting the configuration of storageMax for storing IPFS repository on peer leader as the highest one. This peer must store all blocks from all nodes due to having the primary user and located in the main data center. Next, we set the configuration of storageMax by select other peers which collect the lowest size of total data in daily and consider which peers have a more stable connection. So, the system will make a sequence to replicate them according to those configurations.

In IPFS network, a file which uploaded will map into an object with fixed size using a combination of hash function and base encoding. An object might be several chunks that called as blocks. SHA256 hash function has applied by IPFS and made the size of each block at most 256KB because IPFS use Rabin fingerprint method for chunking files. Rabin fingerprint method use content-defined chunking that identifying each fingerprint based on 20-byte SHA hash value [14]. The content calculation is used to create a hash name for each block, so no duplication occurs with the same content stored at the node. Each file that uploaded to IPFS has each hash name as a Content Identifier (CID) in the network. The data structure in this system use Directed Acyclic Graph (DAG) forming a Merkle tree based on CID that interconnect object as hashes link [6].

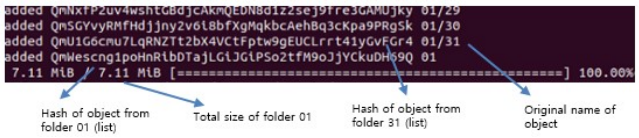


Fig 5. Example of upload data in system

In figure 5, we can see the example of upload the January 2019 data (folder 01) that contain sequence folder of date from 01-31 and each of date folder containing files with size below than 256KB. Each folder has a distinct hash name, but it is connected by a hash link (see fig 6).

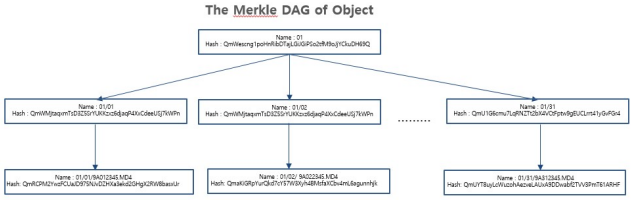


Fig 6. The merkle link of folder 01 from CADI Instrument

If a block has size more than 256KB, it breaks into several chunks of the block according to its total size. Figure 7 depicts the example of a file that has a size 2.2MB. All blocks which derive from the root of the block have a different hash name, and it connected each other as a Merkle tree of the block.

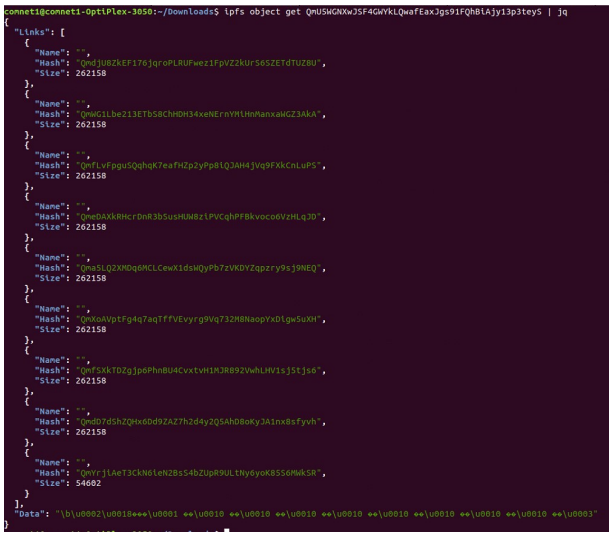


Fig 7. Example of Link of hash from one block that has size more than 2.2MB (20190101_080000.JPG)

This system relies on DHT and BitSwap protocol for block exchanging process. DHT has a job that maps keys to peers on the network and BitSwap who send the block that peers requested. Kademlia DHT is used to store the <key and value> pair of the peer. The value might be as a block hash (if size <= 1KB) or reference of object hash whereas the key is the nodeID who has the block. In this system, the DHT provides two important purposes [7]:

- Peer routing, it is used to find out other peers on the network based on PeerID (nodeID). This information only lists the online node to be announced to others.
- Content routing, it serves information about data published by nodes and records file locations.

When node uploads the latest file to IPFS, it will be broken into blocks and given the hash as its name (CID). If the block size greater than 1 KB, the system records its reference as a value in the DHT. Otherwise, it stores directly in the DHT and DHT is updated. Since this system forces node to share file automatically, at the same time, other peers taking a role as a downloader to get the data and store it locally.

IV. File Retrieval

Peer querying to DHT based on requested hash and system will check in the DHT either it is reference or no. If no, these peers will get directly from DHT, and the system will update the DHT. Otherwise, DHT will route the requester and connect to the peer who stored the hash (*provider*) to get the hash from the provider. Finally, the system stores new records in the DHT.

The central exchange of block on this system utilizes BitSwap protocol. This protocol is a message-based protocol responsible for exchanging block between peers and handles the request to fetch data block from the network. BitSwap manages two processes, first is acquiring a set of blocks (want_list) that have been requested by peer and second is sending a set of blocks (*have_list*) that peer request. After DHT assist a peer in finding another who has the block, P2P connection is built, that peer (sender) sends a message to request the block as its *want_list*. Another peer who listens to the request (*listener*) will check its *have_list* of the block. If there is a match between *want_list* and *have_list*, the block is sent by the listener to the sender directly. BitSwap Ledger records this transaction, and P2P connection is closed. Otherwise, if there is no match between sender and listener, no transaction occurred, no new record in BitSwap Ledger and P2P connection closed.

All process will continue running each time new file created by the instrument. If one of the peers is failures, the data still can be retrieved from other peers. This system has no single point of failure, increase data availability, deduplication and keep the integrity of data.

V. Conclusion

In this paper, we proposed a new system in order to increase data availability for supporting SWIFtS. Our proposal leverages the IPFS network to distribute data and store in IPFS node. We add filesystem watcher to force the node to upload the data automatically and utilize IPFS-Cluster to replicate the data as well as limit the distribution process only in cluster peer members.

In our future works, we need a system monitoring to observe the object flow and find out peer status in real-time. In addition, evaluation should be conducted

to know the system performance such calculate of mean upload/download time and mean throughput.

ACKNOWLEDGMENT

This work was supported by the Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2018-0-00539, Development of Blockchain Transaction Monitoring and Analysis Technology–)

References

[1] Wang, S., Zhang, Y., & Zhang, Y. A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. IEEE Access, 6, 38437-38450. 2018.

[2] Wennergren, O., Vidhall, M., Sörensen, J., & Steinhauer, J. Transparency Analysis of Distributed File Systems: Bachelor Degree Project in Information Technology. University of Skövde, Sweden. 2018.

[3] Benet, J. IPFS-content addressed, versioned, P2P file system. arXiv preprint arXiv:1407.3561. 2014.

[4] Brisbane, S. Decentralising Big Data Processing (Bachelor thesis, The University of New South Wales). 2016.

[5] P. Labs. "IPFS Cluster". 2019. (https://cluster.ipfs.io/).

[6] P. Labs. "IPLD". 2019. (https://ipld.io/).

[7] P. Labs. "IPFS Architecture". 2019. (https://github.com/ipfs/specs/tree/master/architecture)

[8] Antunes, J."Scaling PubSub over the Distributed Web". 2018.(http://web.tecnico.ulisboa.pt/~ist14191/papers/TR-75993-joao-antunes.pdf).

[9] Confais, B., Lebre, A., & Parrein, B. An object store service for a Fog/Edge Computing infrastructure based on ipfs and a scale-out NAS. In *Fog and Edge Computing (ICFEC),* 2017 IEEE 1st International Conference, pp. 41-50. 2017.

[10] Vorick, D., & Champine, L. "Sia: Simple decentralized storage". 2014. (https://sia. tech/sia.pdf).

[11] Wilkinson, S., Boshevski, T., Brandoff, J., & Buterin, V. "Storj a peer-to-peer cloud storage network". 2014.

[12] Lambert, N., Ma, Q., Irvine, D. "Safecoin: The Decentralized Storage Token". 2015. (https://docs.maidsafe.net/Whitepapers/pdf/Safecoin.pdf

[13] Nygaard, R. Distributed Storage with Strong Data Integrity based on Blockchain Mechanisms (Master's thesis, University of Stavanger, Norway). 2018.

[14] Kaiser, J., Meister, D., Brinkmann, A., & Effert, S. Design of an exact data deduplication cluster. In 2012 IEEE 28th Symposium on Mass Storage Systems and Technologies (MSST), pp. 1-12. 2012.

[15] Borthakur, D. HDFS Architecture Guide. Hadoop Apache Project, 53, 1-13. 2008.

컨테이너 기반 M-CORD 모니터링 시스템 설계 및 구현

홍지범°*, 김우중*, 유재형†, 홍원기*

*포항공과대학교 컴퓨터공학과
†포항공과대학교 정보통신대학원

{hosewq, woojoong, styoo, jwkhong}@postech.ac.kr

Design and Implementation of Container-based M-CORD Monitoring System

Jibum Hong°*, Woojoong Kim*, Jae-Hyoung Yoo†, James Won-Ki Hong*

*Department of Computer Science and Engineering, POSTECH
†Graduate School of Information Technology, POSTECH

요 약

Mobile CORD(M-CORD)는 통신사업자가 전화국에서 운용하는 모바일 LTE 네트워크를 엣지 클라우드(edge cloud)의 형태로 서비스하기 위한 플랫폼으로, 이 플랫폼 내에는 LTE EPC 네트워크를 구성하는 다양한 컨테이너(container)들이 구동되고 있다. 하지만 현재 이러한 M-CORD 플랫폼 상의 컨테이너들이 송/수신하는 데이터 트래픽 및 각 컨테이너들의 컴퓨팅 및 메모리 자원을 모니터링하는 시스템이 부재하다는 문제가 있다. 본 논문에서는 M-CORD 플랫폼을 효율적으로 관리 및 운용하기 위한 모니터링 시스템을 설계하고 구현한다. 이를 통해 M-CORD 플랫폼 상의 컨테이너들이 얼마나 많은 컴퓨팅 자원을 소모하고, 얼마나 많은 데이터 트래픽이 송/수신되는지를 모니터링할 수 있다.

Ⅰ. 서 론

기존 네트워크 환경에서 제공되고 있던 서비스들은 최근 하드웨어 및 소프트웨어 자원 활용을 최적화하고, 서비스를 보다 유연하고 효율적으로 관리하기 위해 SDN(Software-Defined Networking), NFV(Network Function Virtualization) 및 클라우드 컴퓨팅(Cloud Computing) 기술과 결합하여 사용자에게 제공되고 있다. 이러한 기술들을 결합한 Central Office Re-architected as a Data Center(CORD)는 통신사업자들이 운용하는 전화국(Central Office) 내에 있는 네트워크 장비 및 기능을 가상화하여 CORD 플랫폼 내 서버에서 동작을 시켜 전화국을 데이터센터로 변경하여 운용할 수 있도록 하는 데이터센터 플랫폼이다.

그중 Mobile CORD (M-CORD)는 기존의 모바일 LTE 네트워크를 CORD 플랫폼을 통해 제공하는 것으로, LTE 네트워크를 서비스하기 위해 기존 LTE EPC(Evolved Packet Core)의 네트워크 기능을 가상화하여 CORD 내 서버에서 구동한다. 이후, 외부에 설치된 기지국(evolved NodeB, eNB)들을 M-CORD 상에서 동작하는 컨테이너로 연결하여 모바일 네트워크를 구축할 수 있다. 또한 M-CORD 는 데이터센터 플랫폼으로 활용할 수 있기 때문에 모바일 엣지 컴퓨팅(Mobile Edge Computing) 및 다양한 서비스들을 제공할 수 있다.

하지만 M-CORD 는 현재 연구 개발이 진행 중인 프로젝트이기 때문에 데이터센터 플랫폼 중 하나임에도 불구하고 M-CORD 상에서 운용되는 다양한 노드 및 컨테이너들의 컴퓨팅 자원이나 네트워크 인터페이스의 처리량과 같은 기본적인 정보조차 모니터링하지 못하는 문제가 있다. 이러한 모니터링 시스템이 부재할 경우, 관리자가 M-CORD 플랫폼을 효율적으로 관리하는데 필요한 자원 최적화 혹은 이상 상태 감지에 따른 의사결정에 어려움을 겪을 수 있다.

본 논문에서는 이러한 모니터링 시스템의 부재 문제를 해결하기 위해 M-CORD 모니터링 시스템에 대한 디자인을 제안하고 이를 구현한다. 그리고 본 모니터링 시스템을 활용한 향후 연구 방향을 제시한다.

Ⅱ. 관련 연구

CORD[1]의 개념적인 구조를 살펴보면 먼저 CORD 플랫폼을 전체적으로 제어 및 관리하는 컨트롤러인 XOS 가 있다. 이 XOS 를 통해 CORD 가 포함하는 구성요소들을 제어 및 관리한다. XOS 아래에 위치하는 구성요소들은 크게 ONOS 컨트롤러, Kubernetes 및 상용 하드웨어로 구성되어 있다. ONOS 컨트롤러는 하드웨어 중 화이트박스(white box) 스위치를 제어하고, 이를 위해 ONOS 컨트롤러에는 스위치의 제어를 위한 어플리케이션(Ctrl App)들이 정의되어 있다. Kubernetes 는 서버 내의 컨테이너를 제어 및 관리하며, 이를 통해 CORD 는 사용자가 원하는 VNF 들을 서버에 배포할 수 있다.

이러한 엣지 클라우드 형태의 데이터센터 플랫폼을 모니터링하기 위해 하드웨어 및 VNF 가 사용하는 자원을 모니터링하는 연구들이 진행되고 있으며, 이를 통해 물리 및 가상 자원에 대한 요구사항과 관련된 문제들을 해결한다[2]. cAdvisor[3]는 컨테이너화된 VNF 에 대한 자원 사용 및 성능 정보를 모니터링하여 제공하고 있으며, Prometheus[4]는 오픈소스 시스템 모니터링 솔루션으로

Node exporter 와 같은 여러 exporter 를 통해 물리 노드의 자원 사용 및 성능 정보를 추출하여 서버 내부의 데이터베이스에 저장하고, 이를 사용자에게 제공한다. 본 논문에서는 이러한 오픈소스를 활용하여 제안하는 모니터링 시스템을 구현하고 이를 가시화하여 제공한다.

Ⅲ. M-CORD Monitoring System

본 논문에서 제안하는 M-CORD 플랫폼 모니터링 시스템은 그림 1 과 같다.

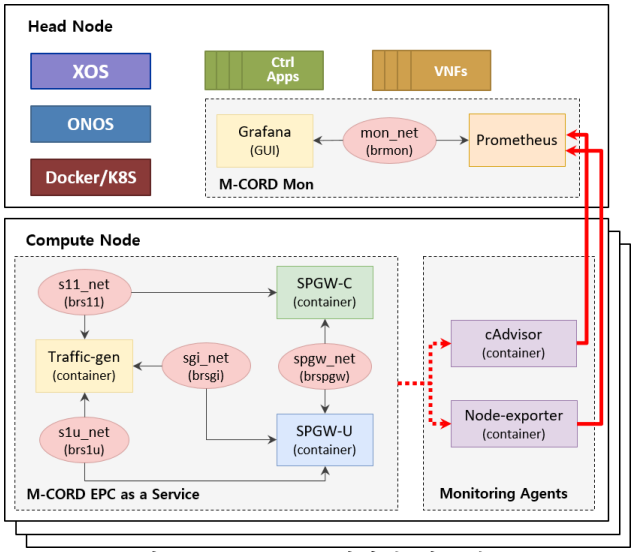


그림 1 M-CORD 모니터링 시스템 구조

M-CORD EPC 네트워크는 3 개의 컨테이너를 가지고 있다. 본 논문에서는 M-CORD 4.1 버전을 사용하였으나 그 이상의 버전에서도 사용이 가능하다. 기존 LTE EPC 네트워크의 구성요소인 Serving Gateway(S-GW)와 Packet Data Network Gateway (P-GW)를 SPGW 로 통합한 후, 제어 평면만 분리한 SPGW-C 와 사용자 평면만 분리한 SPGW-U 로 구성되어 있다. 그리고 SPGW-C 와 SPGW-U 를 검증하기 위한 eNB 에뮬레이터(Traffic-gen)가 존재한다. 이 에뮬레이터는 MME, HSS, eNB, 그리고 인터넷을 에뮬레이션하도록 설계되었으며, NG4T 사의 유료 에뮬레이터 대신 TCPreplay 기반의 컨테이너로 제작하여 사용한다. EPC 네트워크를 구성하는 각 컨테이너들은 컨테이너 기반 가상화 도구인 Docker (18.06 버전)를 사용하고, 컨테이너들 사이의 연결은 4 개의 가상 네트워크 인터페이스로 구성된다.

이와 같은 M-CORD 플랫폼 상의 EPC 네트워크 구성요소를 모니터링하기 위해 본 논문에서는 세 가지의 모듈을 정의하고 구현한다. 먼저 물리 서버 및 컨테이너 내부를 모니터링하는 에이전트(agent) 모듈은 M-CORD 플랫폼이 설치된 서버 내에서 동작을 하며, 물리 서버의 정보를 모니터링하기 위해 Node exporter(0.14 버전)와 컨테이너들의 정보를 모니터링하기 위한 cAdvisor (0.33 버전)로 구현된다. 각 에이전트 모듈은 컨테이너 형태로 동작하며 5 초마다 물리 서버와 컨테이너의 정보를 얻는다. Prometheus(1.7 버전)는 이러한 모니터링 정보를 받기 위해 cAdvisor 와 Node exporter 의 주소를 지정한다. 이후 Prometheus 는 주기적으로 HTTP 엔드포인트에 접속하여 모니터링 정보를 받아 물리 서버 및 컨테이너 별로 구분하여 저장한다. 마지막으로 Web GUI 대시보드는 Grafana(4.4.3 버전)로 구현되며, Prometheus 내부 시계열 데이터베이스(TSDB)에서 저장하고 있는 모니터링 정보를 관리자에게 그래프와 같은 시각화 형태로 제공한다.

그림 2 M-CORD 모니터링 시스템(Web UI)

제안하는 구조를 바탕으로 구현한 M-CORD 모니터링 시스템을 통해 실시간으로 관리자에게 제공되는 Web UI 는 그림 2 와 같다. 먼저 각 물리 서버 별 자원(CPU, 메모리, 디스크 I/O 등) 사용 현황 및 네트워크 인터페이스에서 송/수신되는 트래픽을 모니터링하여 제공하고, 물리 서버 내에서 동작하는 각 EPC 구성요소들의 자원 및 가상 네트워크 인터페이스에 송/수신되는 트래픽을 보여준다. 이 외에도 쿼리를 통해 관리자가 원하는 모니터링 정보를 얻어 그래프, 테이블, 게이지 등의 원하는 형태로 시각화하는 것도 가능하다.

Ⅳ. 결론 및 향후 연구

본 논문에서는 M-CORD 플랫폼에서 현재 부재하는 모니터링 시스템 제안 및 구현하였다. 이를 활용하여 기존 M-CORD 에서 사용 중인 다양한 노드들의 CPU, 메모리 및 디스크 자원에 대한 사용량은 물론, 각 컨테이너들의 자원 사용량과 트래픽 송/수신 상태를 모니터링할 수 있다. 향후 연구로 다양한 유스케이스를 통해 본 M-CORD 모니터링 시스템을 검증하고, 모니터링 정보를 기계 학습을 통해 학습시켜 M-CORD 상에서 동작 중인 컨테이너들에 대한 비정상적인 행동 및 상태를 탐지(Anomaly Detection)하는 서비스 대해 연구할 예정이다.

ACKNOWLEDGMENT

이 논문은 2019 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2015-0-00575, 글로벌 SDN/NFV 공개소프트웨어 핵심 모듈/기능 개발, IITP-2019-2017-0-01633*, 대학 ICT 연구센터지원사업).

참 고 문 헌

[1] L. Peterson, et al., "Central office re-architected as a data center," IEEE Communications Magazine, vol. 54, no. 10, pp. 96-101, Oct. 2016.

[2] K. Fatema, et al., "A survey of Cloud monitoring tools: Taxonomy, capabilities and objectives," Journal of Parallel and Distributed Computing, vol. 74, no. 10, pp. 2918-2933, Oct. 2014.

[3] google, “cAdvisor”, 2019. [Online] (https://github.com/google/cadvisor).

[4] Prometheus, “Prometheus”, 2019. [Online] (https://prometheus.io).

CUDA 다중 프로세스 실행 서비스를 이용한 과학 응용 실행 패턴 분석

*김세진, 오지선, 김윤희
숙명여자대학교 컴퓨터과학과
{*wonder960702, jsoh8088}@gmail.com, yulan@sookmyung.ac.kr

Execution Pattern Analysis of Scientific Applications using CUDA Multi-Process Service

*Sejin Kim, Jisun Oh, Yoonhee Kim
Dept. of Computer Science, Sookmyung Women's University

요 약

Graphical Processing Units(GPUs)는 연산을 병렬적으로 처리함으로써 높은 성능을 사용자에게 제공해왔다. 기술의 발전에 따라 GPU의 하드웨어 자원이 증가함으로써, 하나의 응용만을 실행하면 GPU 자원을 모두 사용하지 못하는 문제가 생겨났다. 이로 인해 새로운 GPU 구조들은 커널의 동시 실행을 지원한다. 하지만 이는 같은 응용 안에서만 동시 실행이 가능하다는 문제점이 있어 NVIDIA는 Multi-Process Service(MPS)를 소개하였다. MPS는 다른 응용에 속한 커널도 동시 실행할 수 있도록 서비스하지만, 응용의 실행 특성이 미리 파악되어 있지 않으면 MPS 장점을 최대한으로 취할 수 없다. 본 논문에서는 응용 프로파일링을 통해 응용의 특성을 파악하고, 응용의 동시 실행을 통해 MPS에서의 커널 실행과 동시에 실행되는 응용의 조합의 중요성을 확인하였다. GPU 커널이 실행되는 비율이 높은 응용들을 동시 실행하여 여러 커널이 중첩 실행되었을 때, MPS를 사용하는 장점을 최대화할 수 있다.

Ⅰ. 서 론

Graphics Processing Units(GPUs)의 사용을 통해 다양한 응용들이 강력한 처리 성능을 받음으로써 여러 분야에서 이를 사용하는 것이 점점 더 보편화 되고 있다. 이는 응용의 연산 집약적인 부분을 병렬적으로 처리함으로써 처리 속도를 높일 수 있다.

특히 NVIDIA GPU의 Compute Unified Device Architecture(CUDA) 프로그래밍 모델은 커널(kernel)을 GPU에서 연산을 수행하기 위해 동작하는 함수로 정의한다. 이는 여러 개의 스레드가 프로그래머의 정의에 따라 스레드 블록을 이루며 각 스레드 블록은 하나의 Streaming Multiprocessor(SM)에서 실행된다. GPU는 수천 개의 코어에서 스레드를 병렬(parallel) 실행하여 응용들을 가속해왔다[5]. 하지만 GPU의 하드웨어 자원이 증가하면서 스레드 병렬성만으로는 GPU 자원들을 충분히 활용하지 못하는 문제점이 발생하였다[1].

이를 위해 최근의 GPU 구조들은 프로세스(응용)의 독립적인 커널들을 다른 스트림으로 지정함으로써 동시 실행(concurrent)을 제공한다. 하지만 이는 같은 프로세스 안에서만 동시 실행이 가능하다는 한계가 있다. 따라서 응용이 충분한 자원을 사용하지 않는다면 GPU의 자원 활용률이 떨어질 수 있다. 이의 한계를 해결하기 위해서 NVIDIA는 Multi-Process Service(MPS)를 제공한다. MPS는 다른 프로세스의 커널을 하나의 프로세스처럼 보이도록 함으로써 동시 수행이 가능하게 한다.

그러나 여러 커널의 실행 및 커널 특성이 미리 파악되어 있지 않으면 MPS의 장점을 최대한으로 활용할 수 없다. 그러므로 응용의 프로파일링을 통해서 응용의 성격을 파악하고 서비스를 이용하는 것이 중요하다. 본 연구에서는 MPS를 사용한 응용의 스케줄링 알고리즘을 개발하기 위해서 MPS를 사용한 CUDA 과학 응용의 실행 패턴을 분석하고자 한다. LAMMPS[3], GROMACS[4] 응용의 동시 실행을 통해 MPS의 커널 수행 방식과 동시에 실행되는 커널 간 조합의 중요성을 확인한다.

본 논문의 구성은 다음과 같다. 2장에서는 MPS의 구성과 기능을 설명한다. 3장에서는 과학 응용인 LAMMPS와 GROMACS의 실행을 통해 MPS의 성능 확인 및 분석을 진행하며, 4장에서는 결론 및 향후 연구에 관해 기술한다.

Ⅱ. Multi-Process Service

CUDA 응용은 사용할 하드웨어 자원이 담긴 CUDA context를 생성하며 시작된다. Hyper-Q 기술은 하나의 커널이 모든 GPU 자원을 사용하지 않는다면, 같은 CUDA context에 속한 다른 커널을 동시에 실행시킬 수 있도록 한다[6]. CUDA 모델에서 스레드 병렬성과 함께 동시에 실행될 수 있는 CUDA 작업은 커널, 디바이스-호스트간 메모리 복사 작업, CPU에서의 작업이다. Hyper-Q 기술을 가진 GPU의 동시 실행 스케줄러(concurrent scheduler)는 이 작업들을 중첩시켜 같은 context에 속한 여러 프로세스가 GPU 자원을 동시에 활용할 수 있도록 한다. 동시 실행 스케줄러는 커널과 디바이스-호스트 간 메모리 복사작업을 가지는 세 개의 작업이 제출되었을 때, 그림 1과 같이 커널과 메모리 복사 작업을 중첩한다. 하지만 다른 context에 속한 커널들은 time-sliced 스케줄러를 통해 그림 1과 같이 순차적으로 수행됨으로써 실행시간이 늘어남을 확인할 수 있다. 그러므로 하나의 context가 충분한 자원을 사용하지 않는다면 GPU의 자원이 낭비될 수 있다.

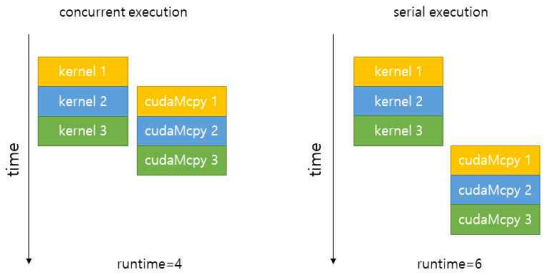


그림 1 동시 실행과 순차 실행

MPS는 다른 CUDA context로부터 온 여러 커널이 MPS 서버를 통해 작업을 제출함으로써 하나의 context로 관리되어 Hyper-Q 기술을 최대한으로 활용할 수 있도록 제공하는 CUDA API이다. GPU 남는 자원을 다른 프로세스의 커널이 사용할 수 있도록 하여 MPI(Message Passing Interface) 작업들을 실행하기에 적합하다[2]. MPS의 구조는 그림 2와 같이 서버 프로세스, 클라이언트 프로세스, MPS 컨트롤 데몬 프로세스로 구성되어 작동한다.

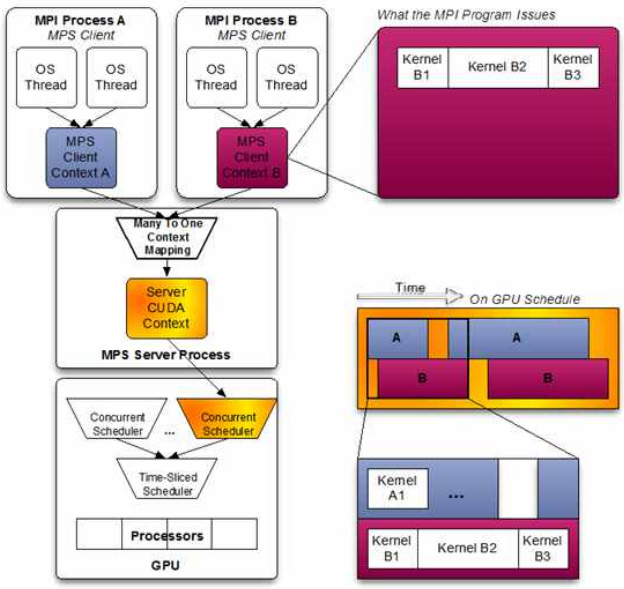


그림2 MPS 클라이언트-서버 구조[2]

단일 프로세스는 GPU에서 사용 가능한 자원을 모두 사용하지 않는다. MPS는 서로 다른 프로세스(응용)의 커널과 호스트와 디바이스간의 메모리 복사 작업을 중첩해서 시간을 단축하고, GPU 활용률을 높일 수 있다. GPU는 각 프로세스에 저장공간과 스케줄링 자원을 할당한다. 따라서 여러 프로세스가 GPU에서 실행되면 스케줄링 자원이 GPU에서 스와핑되어야 한다. 하지만 MPS 서버는 모든 MPS 클라이언트가 GPU의 저장공간과 스케줄링 자원을 공유하게 하므로 클라이언트를 스케줄링할 때 스와핑 오버헤드를 제거한다.

Ⅲ. 실험 및 분석

본 논문에서는 MPS의 프로세스 실행 방식의 진행과 성능을 확인 위해 과학 계산 응용인 LAMMPS[3], GROMACS[4] 응용을 대상으로 실행하였다.

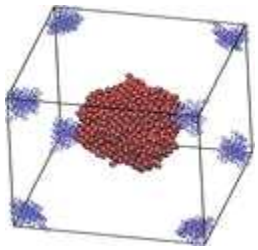


그림 3-(a) LAMMPS[3]

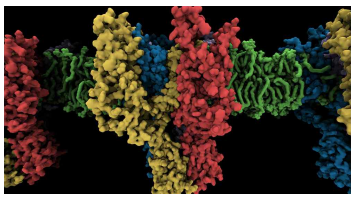


그림 3-(b) GROMACS[7]

그림 3-(a)와 같이 LAMMPS 응용은 분자 동역학 시뮬레이션을 위한 소프트웨어이며, 그림 3-(b)와 같이 GROMACS는 물 분자 데이터 셋을 사용하여 뉴턴의 운동 방정식을 시뮬레이션 한다. 본 실험 환경은 Intel(R) Core(TM) i7-5820K, Nvidia Titan XP이며 Pascal 구조이다. 각 응용 프로세스의 커널 실행 흐름을 파악하기 위해 Nvidia visual profiler(nvprof)를 사용하여 확인하였다.

LAMMPS 응용의 총 수행시간은 1196초이며 수행 동안 363MB~8.3GB의 메모리를 사용한다. 이 응용은 호스트와 디바이스간 메모리 복사가 빈번하게 일어나서 GPU 커널이 실행되는 비율을 나타내는 GPU의 평균 활용도(utilization)가 약 25%이다. GROMACS 응용의 경우, 총 수행시간은 636초이며 수행 동안 171MB~537MB의 메모리를 사용한다. 이 응용은 수행시간 동안 빈번히 연산이 실행되어 평균 활용도(utilization)가 69%를 보이며 계산 집약적인 형태를 확인할 수 있다.

MPS를 사용하여 LAMMPS와 GROMACS를 단독적으로 수행하였을 때 수행시간은 각각 1208초, 641초였다. MPS를 사용하지 않았을 때와 비교하여 수행시간이 각각 12초, 5초 늘어난 것을 확인하였다. 이는 MPS 서버와 클라이언트 간의 통신 오버헤드로 인하여 수행시간이 증가한 것임을 알 수 있다.

MPS를 사용한 응용의 동시 실행을 통해 MPS의 커널 수행 방식을 확인하기 위해, 동시 작업을 LAMMPS-GROMACS, GROMACS-GROMACS로 구성하여 실험하였다. LAMMPS-GROMACS 작업의 경우 수행시간은 MPS를 사용하지 않고 동시에 실행했을 때의 시간은 833초, MPS를 사용하였을 때의 시간은 809초이다. 이때, 본 논문은 커널이 동시에 실행된 경우의 커널 수행 방식에 초점을 두고 있으므로 두 응용이 동시에 실행되는 시간만 고려하였다. 두 응용의 수행시간이 다르므로 하나의 응용의 수행이 끝나고 다른 응용이 단독으로 수행되는 시간은 고려하지 않았다.

그림 4-(a)는 MPS를 사용하지 않았을 경우의 LAMMPS 응용과 GROMACS 응용의 커널 실행(파란색 계열 작업)과 CPU-GPU 메모리 전송 작업(황토색 작업)을 343초부터 344.5초까지 보여주는 타임라인이다. 두 프로세스는 다른 CUDA context를 생성하여 스케줄링 자원을 할당한다. 343.7초부터 343.9초까지 LAMMPS의 메모리 작업이 실행되는 동안, GROMACS의 커널이 약 155밀리 초 중첩되는 등 GPU의 효율성을 높일 수 있었다. 그러나 343.1초, 343.4초에는 커널과 메모리 전송 작업이 중첩되지 않고, 각 응용의 커널들이 함께 제출된다. 이때. time sliced 스케줄러가 커널들을 시간 단위로 나누어 순차적으로 실행시키므로 문맥교환의 오버헤드가 발생하며 GPU 자원 사용의 효율성이 낮다.

파스칼 구조의 NVIDIA GPU는 명령어 레벨로 문맥 교환이 이루어지기 때문에 블록 단위로 보여주는 nvprof에서는 커널의 문맥 교환을 확인할 수 없다.

그림 4-(b)는 325초부터 326.75초까지 각 응용이 MPS를 사용하여 하나의 CUDA context로 통합되어 두 개의 stream으로 실행되는 타임라인이다. 325.4초부터 325.6초까지의 LAMMPS 메모리 복사 작업이 약 141 밀리 초 동안 GROMACS의 커널과 중첩될 뿐 아니라, 325.7초부터 약 0.1초 간격으로 약 326초까지 LAMMPS의 연산 작업과 GROMACS의 메모리 전송 작업이 중첩되어 효율성을 높일 수 있었다. 또한, GROMACS의 커널에서 LAMMPS의 커널로 문맥이 변경될 때, 스케줄링 자원이 GPU에서 스왑 인, 아웃 되지 않으므로 문맥 교환의 오버헤드를 제거할 수 있다.

MPS를 사용하지만 수행시간에서의 차이는 거의 발생하지 않는데, 이는 LAMMPS와 GROMACS 응용의 커널이 동시에 실행되는 구간이 많이 존재하지 않아 문맥 교환의 오버헤드가 많이 발생하지 않기 때문이다. 또한, 이 두 응용은 MPS를 사용하지 않아도 커널과 CPU-GPU 메모리 복사 작업을 중첩하여 수행시간이 짧고 GPU 활용도가 높다. GROMACS-GROMACS 작업의 경우 MPS를 사용하지 않고 실행했을 때 1391초, MPS를 사용하였을 때는 1233초 수행시간이 소요된다.

그림 4-(c)는 MPS를 사용하지 않았을 때 LAMMPS-GROMACS 실행을 179.5초부터 180.15초까지 프로파일링한 타임라인이다. 두 프로세스는 다른 CUDA context에 속하여 실행된다. 커널(하늘색 작업)들이 동시에 제출이 되어 time sliced 스케줄러가 시간을 분할하여 순차실행이 되고, 문맥이 교환될 때 스케줄링 자원이 GPU에서 스와핑된다. 호스트-디바이스간 메모리 작업(황토색 작업)과 커널이 중첩되어 실행되기보다는, 커널이 동시 실행되므로 GPU 자원을 두고 경쟁이 일어나 효율성이 떨어지며 수행시간도 길어진다.

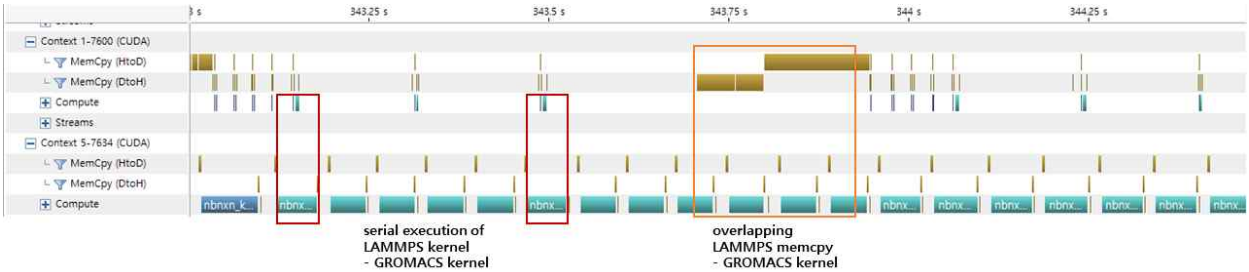


그림 4-(a) MPS를 사용하지 않은 LAMMPS-GROMACS 작업의 수행 프로파일

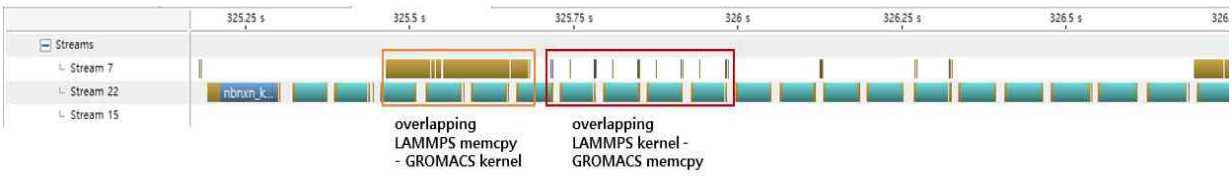


그림 4-(b) MPS를 사용한 LAMMPS-GROMACS 작업의 수행 프로파일

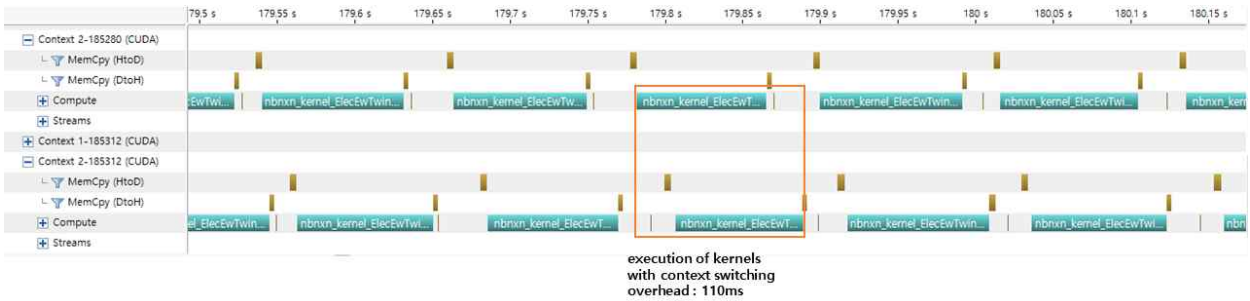


그림 4-(c) MPS를 사용하지 않은 GROMACS-GROMACS 작업의 수행 프로파일

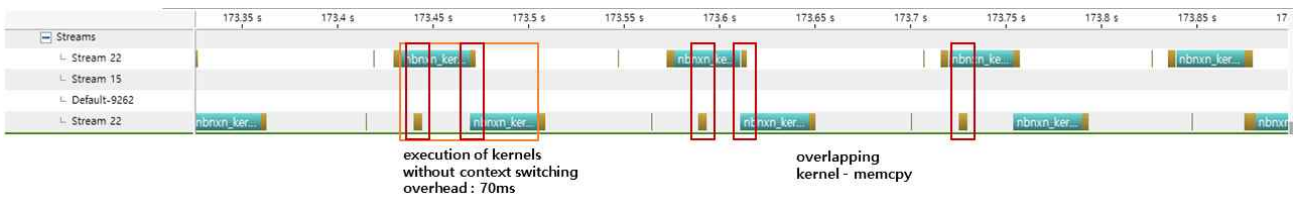


그림 4-(d) MPS를 사용한 GROMACS-GROMACS 작업의 수행 프로파일

그림 4-(d)는 MPS를 통해 두 응용을 수행했을 때 173.3초부터 173.9초까지의 타임라인을 나타낸다. 173.43초부터 약 0.15초 간격으로 두 번째로 제출한 프로세스의 메모리 복사 작업과 첫 번째로 제출한 프로세스의 커널이 중첩되며, 173.46초부터 약 0.15초 간격으로 첫 번째 프로세스의

메모리 복사 작업과 두 번째 프로세스의 커널이 중첩되는 것을 확인할 수 있다. 또한, MPS를 사용하여 각 프로세스의 두 개의 커널을 실행하였을 때는 70밀리 초가 소요되었지만, MPS를 사용하지 않았을 때는 같은 커널의 실행에 대해 110밀리 초가 소요되었다. 이를 통해 MPS는 스케줄링 자원의 GPU 스와핑을 제거하여 문맥 교환의 오버헤드를 축소하고, 커널과 메모리 복사 작업을 중첩하여 수행시간을 단축할 수 있다는 것을 알 수 있다. 따라서 MPS 서버 사용의 장점은 여러 커널이 중첩되는 프로세스일 때 나타나는 것으로 확인할 수 있다.

Ⅳ. 결 론

MPS를 사용해서 여러 응용 및 프로세스들을 동시에 수행할 수 있으며 Hyper-Q 기술을 최대한 활용할 수 있다. 그러므로 이는 GPU 자원 활용률을 높이고, 시간을 단축시킬 수 있다. 본 논문에서 진행한 실험을 통해 여러 커널이 중첩되어 사용되었을 때 MPS를 사용하는 장점을 얻을 수 있다는 것을 확인할 수 있었다. 응용 특성을 파악하여 실험하게 되면 MPS 사용의 장점을 극대화 시킬 수 있으므로 응용의 프로파일링 정보를 미리 파악하는 것이 중요하다.

향후 연구로는 GPU CUDA 응용의 커널 실행 패턴을 파악하여, 프로파일링 정보를 바탕으로 MPS 서버에서 서로 영향을 주는 응용들의 커널 스케줄링 기법에 관해 연구하고자 한다.

ACKNOWLEDGMENT

이 논문은 2015년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2015M3C4A7065646)

참 고 문 헌

[1] Carvalho, Pablo, et al. "Kernel concurrency opportunities based on GPU benchmarks characterization." Cluster Computing (2019): 1-12.

[2] Multi-Process Service, https://docs.nvidia.com/deploy/mps/index.html

[3] LAMMPS, https://lammps.sandia.gov/.

[4] GROMACS, http://www.gromacs.org/

[5] CUDA C Programming Guide, https://docs.nvidia.com/cuda/cuda-c-programming-guide/

[6] HyperQ, http://developer.download.nvidia.com/compute/DevZone/C/html_x64/6_Advanced/simpleHyperQ/doc/HyperQ.pdf

[7] GPU-accelerated GROMACS, https://www.nvidia.com/en-us/data-center/gpu-accelerated-applications/gromacs/

VNF 를 위한 자동화된 스케일링 응용과 마이크로서비스 기반 모니터링

아시프 매흐무드, 송왕철*
제주 대학교
asif@jejunu.ac.kr, *philo@jejunu.ac.kr

Autoscaling application and microservice based monitoring for VNFs

Asif Mehmood, Wang-Cheol Song*
Jeju National University
asif@jejunu.ac.kr, *philo@jejunu.ac.kr

Abstract

Orchestration and monitoring of network resources is an important part in 5G. As the performance of VNF orchestration is dependent on monitoring [1], so the monitoring performance should be better. So that a vast range of decision-making processes such as autoscaling [2] can be done with full ease, solely based on the real-time data. 5G pushes the network towards virtualization, which imposes a limitation of adding an extra layer to the network, indirectly leading to an increase in the processing latency. The proposal aims to solve this problem by adding an autoscaling application which takes real-time data through a custom monitoring microservice application which itself communicates with the slightly modified management microservices instead of relying on a cloud-based monitoring service to reduce the processing latency. This mechanism reduces the process latency because of efficient monitoring, thus providing a better prerequisite for the autoscaling application.

Ⅰ. Introduction

As the networks are evolving everyday by the invention and research of existing as well as the newer technologies. The same kind of an open source framework provided by ONF. The framework provides us to deploy our network services solution inside the data center. The framework by using its extra layer of orchestration [3] above the two integrated projects OpenStack and ONOS, which provides an NFV architecture.

It has XOS, OpenStack and ONOS. XOS has different services that provide different rich features for the business entities to indirectly interact with cloud and sdn services and to customize them by their deployment/configuration. The architectural differences are discussed inside the proposed system section. As a lot of systems of which one is a monitoring system [1] contains dependencies on the cloud and sdn platforms. Our system covers this lackness as a solution. with a solution having well defined interfaces with justifications

After this literature, in order to provide a detailed description, necessary background knowledge is provided in comparison. Then the proposed system is discussed by comparing its architectural advantages over the other systems one by one.

Ⅱ. Literature review

SDN & NFV are two of the main key terms that ae used for future networks. The monitoring [1] of these virtual resources inside the NFV architecture implementation sections need to be carefully designed, so that it does not take up the all resources to monitor and it also does not take too much time for the processing. In order to make the process less latent and efficient, the microservice based architecture was followed inside a virtualized datacenter environment. Open source developed network service were used for testing purposes.

In the paper “Auto scaling of data plane VNFs in 5G networks” author proposed a solution with a discussion, how to scale the data plane [2] resources. So, we follow the same principles to achieve scaling mechanism but we are proposing it for the control plane functions as well. It is for all of the VNFs inside the composed network-service.

The concept of [4] network slicing [4] in order to create traffic isolation from other services provided by the same network operator or by another network operator. We take care of the resource isolation while autoscaling mechanism being applied. So, this paper covers the implementation of network slicing which we take as a prerequisite to slice the network.

* Corresponding Author

Author of the paper "Implementation of VNFC Monitoring Driver in the NFV Architecture" describes the necessity of a monitoring drive to be placed inside the elementary management services or closer to where the management services are deployed. As it this approach ends in an environment where the process latency is very low causing the system to behave in an efficient manner.

M-CORD [5] is an open source framework which provides a very way to provision everything as a service inside the data centers, whether they be the mobile, residential or an enterprise data center. It provides a flexible environment to create an end to end connection between the services you create and deploy. The services dependencies can be defined inside the TOSCA (Topology and Orchestration [3] for Cloud Applications) format. In turn, this platform uses OpenStack (cloud) and ONOS (sdn) to fulfil and translate those TOSCA requirements into a final result intended by the user. The user can interact with the XOS in order to fulfil the further needs which can be fulfilled at runtime due to some dependencies not letting the user to achieve all of the goals.

OAISM is an organization that provides open source network services following the standards of NFV. The network services such as eNodeB, vEPC (comprising of vMME, vHSS, vSPGW-C, vSPGW-U) are provided by the above-mentioned organization. eNodeB as a simulator for traffic generation and the rest of the parts are the core network services of LTE architecture.

Ⅲ. Proposed System

The system we propose needs attention to solve the foreseen problems in the field of networks. As the network resources are being moved to the virtual resources and their capabilities of processing are different than the physical machines. So, in order to automate the mechanism in an effective way, the automated decisions must be much closer to reality.

The proposal consists of two contributions. First one is an autoscaling [2] application shown in **Fig. 1**. This application consists of an autoscale controller, data-handler and a configurator. Each of these modules are important, forming the basis for autoscale decision making. The controller is the brain for deciding how much resources to be scaled up/down. And then these configurations are passed onto a configurator which supports the translation of those decisions made by our controller to a specific format or allowing the application developer to configure them via REST APIs provided by an underlying orchestrator. The content or data used for decision making of the algorithm is provided by our proposed and developed monitoring microservice application. The data fetched from the management layer is stored inside our proposed data-handler. This way the resources in the form of VNFs can be scaled automatically, upon different situations.

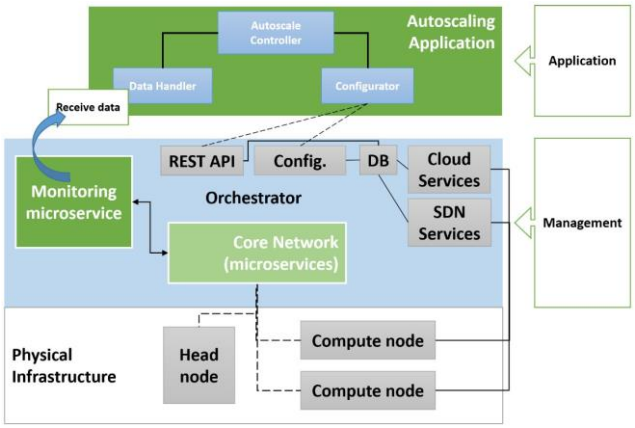


Fig. 1: Overall Architecture

Fig. 2 illustrates the second part we propose i.e. monitoring microservice. The monitoring part, which is different from the OpenStack or SDN monitoring applications in the sense that it is custom and is less latent as compared to them. As a reference, we can think of it as a microservices inside a virtualized data center. The monitoring microservice is directly connected to each VNF's (whether a container or a virtual machine) elementary services. And in the traditional case, this monitoring application is deployed on cloud infrastructure which meant that we cannot know where is the monitoring application located and where does the VNF lie. Hence, in the case of a monitoring service deployed by cloud infrastructure, one cannot assume how much the processing latency can be. In most cases, it is relatively increased. So, in order to avoid the extra service communication done by OpenStack services among them causing the processing latency. In other words, our proposed architecture containing the monitoring microservice is directly connected to the VNF management service, and as a result it reduces the process latency between the services and increases the performance of the overall system.

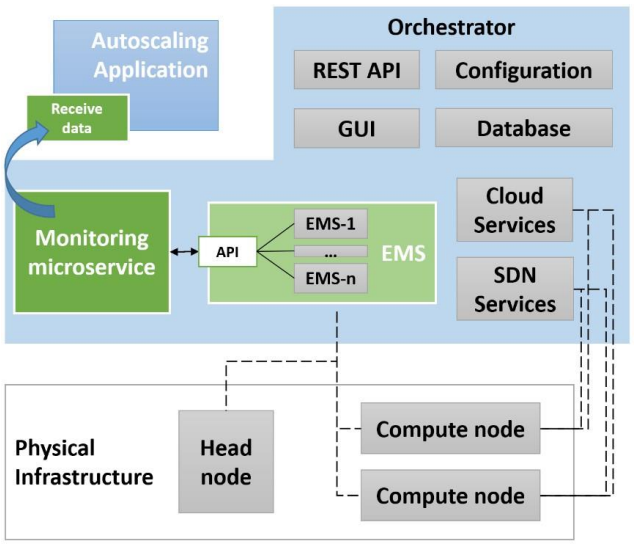


Fig. 2: Monitoring microservice details

Another important aspect of the application is that the default microservices/synchronizers for each of

the VNF inside the topology needs a slighter modification. That is to expose an API interface, proposing a platform and language independent communication mechanism using the JSON format exchanged between the senders/receivers.

So, this way the communication can be done with platform independency and let the application developers focus on the logic and what to take than to take care of the platform. This provides freedom to the software service providers to focus on what to develop. The same case

In order to dump the real-time data, a simple store was made where the dummy data was store in order to generate data for future use. The main goal is to use this data later for research purpose, helping out the real-time environments to take decisions where the situations are work critical. Allowing the application developers to use the real-time data easily with better performance.

An example is given in **Fig. 3**, where it shows to the internal mechanism between the monitoring system [1] and the API interfaces exposed by the elementary management services. The example also shows the detail the EPC elementary services which is itself a composite (a combination of multiple services) service having sub-services such as vMME, vHSS, vSPGW-C and vSPGW-U. It solely communicates on those interfaces by passing messages in JSON format.

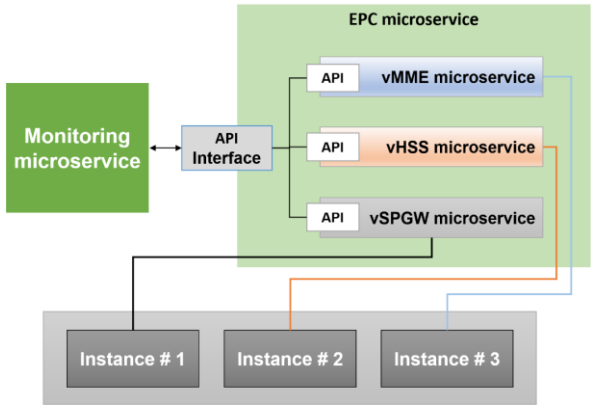


Fig. 3: An example of monitoring microservice communication with VNF microservices

The mentioned architecture shows inner details of an LTE architecture for providing network services. The details for composite services can be seen that how it eases the process of monitoring. The cloud and sdn services that are not supposed to intervene in the process of monitoring do not do extra processing. As a result, the other information such as number of instances as well as the other necessary information. This way by forcing the architecture, the data driven model allows to restrict the interaction to specific data. So, in almost all of the discussions made above, this model is better, feasible and closer to what reality expects rather than relying on the other cloud and sdn services, which do unnecessary processing.

Ⅳ. Conclusion

Though the autoscaling [2] is a mandatory part of the orchestration [3] specially in the future networks and other systems alike, and is very simple but is as important for the application developers. This proposal provides an architecture which has less latency for the monitoring service and the autoscaling application (in NFV environment) can take decisions on the basis of the real-time data. This way the our autoscaling mechanism does not cause any network service disruption, which makes the process of scaling a smooth process in NFV environment which is composed of an orchestrator, elementary services, and NFVI(s). This mechanism introduced and proposed lets the application developers focus on the development rather than to think about the platform and dependencies.

ACKNOWLEDGMENT

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2019-2017-0-01633) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

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] Hyunsik Yang. “Implementation of VNFC Monitoring Driver in the NFV Architecture”, 2017.

[2] Tulja Vamshi Kiran Buyakar. “Auto scaling of data plane VNFs in 5G networks”, 2017

[3] https://wiki.openstack.org/wiki/Tacker

[4] Mehmood Asif. “Network Slicing for 5G Architecture using M-CORD”, May 2018.

[5] M-CORD. https://www.opennetworking.org/m-cord

[6] Asif Mehmood. “An intent-based mechanism to create a network slice using contracts”, 2019.

인공지능 기반 NFV 관리 시스템 구조 및 테스트베드 구축

정세연 [1], 이도영 [1], 유재형 [2], 홍원기 [1]

[1] 포항공과대학교 컴퓨터공학과

[2] 포항공과대학교 정보통신대학원

{jsy0906, dylee90, styoo, jwkhong}@postech.ac.kr

Design of AI-based NFV Management System and Testbed Construction

Seyeon Jeong[1], Doyoung Lee[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

가상화(virtualization) 기술의 발전과 더불어 NFV(Network Function Virtualization) 기술이 도입됨에 따라, 기존 하드웨어 및 제조사 중심의 네트워크 대비 보다 유연하고 기민하며 비용효율적으로 (가상) 네트워크 기능을 운영할 수 있을 것으로 기대된다. 그러나 통신사업자의 NFV 환경에서는 엣지 및 코어 네트워크에 걸쳐 수천~수만 개 이상의 가상 머신(컨테이너) 및 가상 스위치 등이 이용될 것으로 예상되어 그에 따른 네트워크 관리 복잡도 또한 전에 없이 크게 증가할 것으로 전망된다. 따라서 최근에 네트워크 관리에 기계학습(machine learning) 기반의 인공지능(artificial intelligence) 기술을 접목하여 관리자의 의사결정을 돕거나 자율적으로 최적화하려는 시도가 큰 주목을 받고 있다. 본 논문에서는 인공지능 기반 NFV 관리 시스템의 구조를 제안하고 이를 실현하기 위한 테스트베드 구축 내용을 소개한다.

Ⅰ. 서 론

NFV(Network Function Virtualization)는 기존 유/무선 네트워크에서 핵심적인 기능을 수행하는 미들박스(middle-box) 기반 주요 네트워크 기능을 소프트웨어로 구현하여, 가상 머신(Virtual Machine) 및 컨테이너(container) 형태로 실행하는 새로운 네트워킹 패러다임을 말한다. 이를 통해 네트워크 사업자 또는 운영자는 하드웨어 장치 및 벤더(vendor) 중심의 기존 네트워크 대비 CAPEX/OPEX 감소, 새로운 서비스의 신속한 도입, 고장 상황에 유연하게 대처하는 agility 와 flexibility 를 가질 수 있을 것으로 전망된다[1].

기존의 미들박스 기반 운용 환경과 달리, 통신사업자의 NFV 환경에서는 수천~수만 개 이상의 가상 머신 및 가상 스위치를 기반으로 네트워크 기능이 수행되므로, 이에 대한 관리 복잡도가 크게 증가하게 된다. 따라서, 잘못된 네트워크 구성/설정에 대한 가능성이 커지며, 그 결과 자원 사용의 비효율성, QoS 저하, 네트워크 고장과 같은 문제가 발생할 수 있다. 따라서 복잡도가 높은 네트워크에서의 효과적이고 자동화된 관리에 대한 필요성이 대두되고 있다.

최근에 인공지능(artificial intelligence) 기반으로 복잡한 네트워크를 관리하려는 시도가 활발히 진행되고 있다. 이는 (모니터링) 데이터의 지도(supervised) 및 비지도(unsupervised) 학습(learning)을 통해 관리사항들을 예측 및 분류하여 사람(관리자)의 판단을 돕거나, 강화학습(reinforcement learning) 등을 통해 최적 결정을 자율적으로 판단하고 적용하는 기능 등을 포함한다[2].

본 논문은 인공지능 기반으로 주요 NFV 관리 기능을 수행하기위한 시스템의 구조를 제안하고, 이를 실현하기 위해 구축된 NFV 테스트베드의 구현 내용을 설명한다.

Ⅱ. 관련 연구

T-NOVA[3]는 OpenStack 기반의 NFV 운용 환경을 구축하여 OpenStack Ceilometer 를 통해 물리 서버 및 VNF(Virtual Network Function)가 동작하는 가상 머신의 리소스를 모니터링하고, 개별 가상 머신에 에이전트(agent)를 배치하여 운영체제 및 VNF 수준의 모니터링 정보를 관리자에게 제공한다. 또한 SDN 컨트롤러로부터 네트워크 정보를 수집한다. 하지만 인공지능 기반 NFV 및 VNF 관리를 위한 고려사항은 포함하지 않는다.

OpenStack Tacker[4]는 OpenStack 기반으로 가상화(virtualization)되어 동작하는 서버 풀(pool)로부터 NFV 환경을 구축, VNF 및 SFC(Service Function Chain) 수준에서 NFV 관리 기능을 제공한다. 일반적으로 관리자의 수동 설정과 카탈로그를 이용한 deployment 를 가정하지만, 다양한 API 를 제공하는 오픈소스이므로 코드 수정을 통해 NFV 관리 인터페이스를 확장시킬 수 있다.

KDN(Knowledge-Defined Networking)[5]은 네트워크 운영 및 제어에 인공지능 기법을 적용하기 위한 구조 및 방법론을 제시한다. 또한, 모니터링 데이터를 학습시킨 기계학습(machine learning) 모델로부터 트래픽 볼륨에 따른 VNF 별 리소스 사용량 예측 등의 유스케이스를 통해 그 실효성을 보인다. 하지만 제안된 KDN 구조를 포괄하는 시스템 수준의 구현 결과물은 알려지지 않고있다.

본 논문은 인공지능 기반 NFV 관리 시스템의 제안

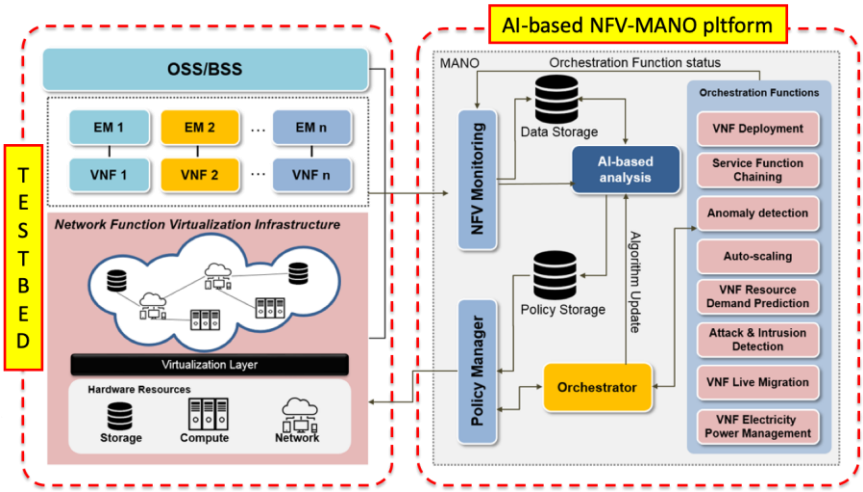


그림 1. 인공지능 기반 NFV 관리 시스템 구조도

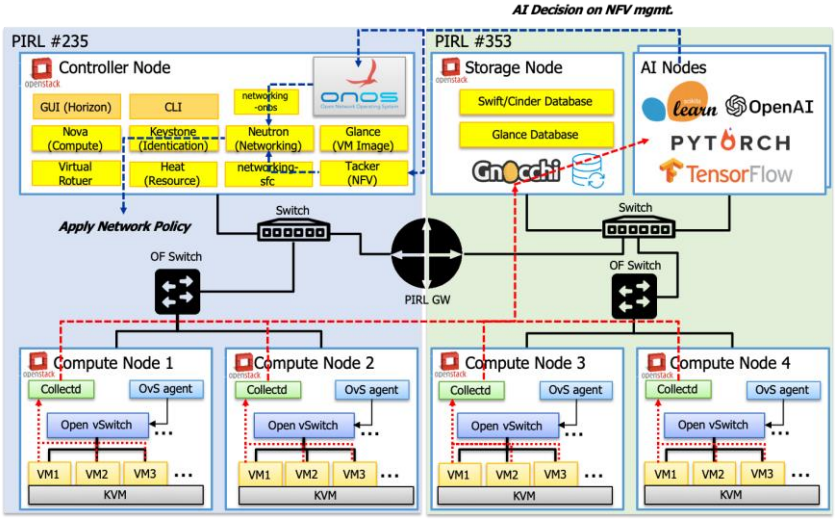


그림 2. NFV 테스트베드 구조도

및 테스트베드 구축을 위해, 상기 연구에서 제시된 구조 및 구현 내용을 참고하고 제약사항을 고려하였다.

Ⅲ. 인공지능 기반 NFV 관리 시스템 제안

그림 1 은 인공지능 기반 NFV 관리를 위한 시스템의 전체적인 구조도를 보인다. 그림 1 의 우측 Orchestration Functions 는 인공지능 기반 주요 NFV 관리 기능(기계학습 모델)을 나타내며, 이를 위해 NFV 모니터링 데이터를 수집하고 전처리하여 전달해주는 NFV Monitoring 모듈이 필요하다. 기계학습을 통한 모니터링 데이터의 분석 결과는 네트워크 관리 정책(policy)의 형태로 변환되어, Policy Manager 를 통해 그림 1 의 좌측에 위치한 기존 VNF 또는 SFC 의 구성 및 deployment 를 변경하거나, Orchestrator 를 통해 리소스를 재분배 시킨다. 또한, 기계학습 기반 정책 결정과정에서 특정 네트워크 서비스의 QoS 최적화 또는 최대 리소스 사용률(utilization) 등의 다중 목표 함수를 설정할 수 있도록 open loop 형태로 동작해야 한다.

아래는 제안된 시스템에서 기계학습 기반 인공지능으로 해결하고자하는 주요 NFV 관리 기능이다 [6].

1. NFV Monitoring: 물리/가상 자원 사용량 및 VNF 성능/장애 상태 감시
2. VNF Deployment: 다중 목적(QoS, 자원효율 등) 고려하여 최적의 서버에 VNF 를 배치
3. Service Function Chaining: 네트워크 서비스의 SLA 보장을 목표로 최적의 QoS 를 가지는 VNF 체이닝 구성
4. Resource Anomaly Detection: VNF 의 비정상적인 자원 사용 탐지
5. Attack & Intrusion Detection: 네트워크 공격에 대한 (실시간) 탐지
6. Auto-scaling: 네트워크 부하나 자원 사용 패턴을 예측하여 VNF scaling-in/out 및 up/down 기능을 선제적으로 수행
7. VNF Live Migration: 장애 가능성을 사전 탐지하고 발생 전에 미리 이동시키며 이동에 따른 단절 시간 최소화

Ⅳ. 테스트베드 구축

본 장에서는 제안된 인공지능 기반 NFV 관리 시스템의 실현을 위해, 선행연구로서 진행된 NFV 테스트베드 구축 내용을 설명한다.

그림 2 는 물리적으로 이격된 교내 서버실 두 곳에 구축한 NFV 테스트베드의 구조도이며, 서버 풀은 OpenStack 을 통해 클러스터링된다. 각 서버실 내 Compute 노드는 OpenFlow 스위치로 연결되며, 각 Compute 노드 내 가상 머신(VM)들은 Open vSwitch 가상 스위치에 연결된다. 이를 통해 ONOS SDN 컨트롤러는 네트워크 상태 및 통계치를 수집하며, NFV 관리 정책에 따른 트래픽 제어를 용이하게 한다. OpenStack 네트워킹에 있어 Neutron 과 ONOS 간의 상호운용성(interoperability)을 위해 networking-onos 플러그인을 사용할 수 있으나, Tacker 에서의 SFC 지원을 위한 networking-sfc 플러그인과의 충돌 문제를 해결하기 위해 추가 구현이 필요하다.

VNF 가 동작하는 가상 머신의 리소스 상태(CPU, 메모리, 디스크 등)는 각 Compute 노드 내 collectd 에이전트를 통해 초 단위로 주기적으로 수집되며, 이 데이터는

Storage 노드 내 Gnocchi 시계열 데이터베이스에 저장된다. 저장된 모니터링 데이터는 AI 노드 내 각 NFV 관리 기능별 기계학습 모델(그림 1)로 전달되어 학습된다. 이 때 서로 다른 형식 및 범위를 가지는 데이터의 전처리(normalization 등) 과정은 추가적인 자동화 구현이 필요하다.

그림 1 에서 각 NFV 관리 기능에 대한 기계학습 모델로 출력되는 네트워크 정책들을 ONOS 와 Tacker 로 전달하여 실제 네트워크에 적용한다. Traffic engineering 정책의 경우 ONOS 의 관련 application 에 의해 처리되어 (가상) 스위치에 반영되며, VNF 및 SFC 의 deployment 에 관련된 정책은 Tacker 의 VNF 및 Network Service Descriptor 내 동적 파라미터 형태로 전달된다(VNF 체이닝 순서, placement 대상 서버 등). 리소스 관련 정책(auto-scaling 등)은 Tacker 의 API 호출로 처리되어 Nova 등에 의해 네트워크에 반영된다.

구축된 NFV 테스트베드 상에서 기계학습 기반 NFV 관리의 적용가능성(applicability)을 검증하기 위해, Tacker 를 통해 VNF 를 배치하고 SFC 로 구성(그림 3)하여 시간에 따라 가변적인 양의 트래픽을 주입하였다. 이를 통해 VNF 별로 수집되는 리소스 사용량 데이터의 기계학습 모델을 생성, 향후 VNF 의 리소스 사용량을 예측하였다[7].

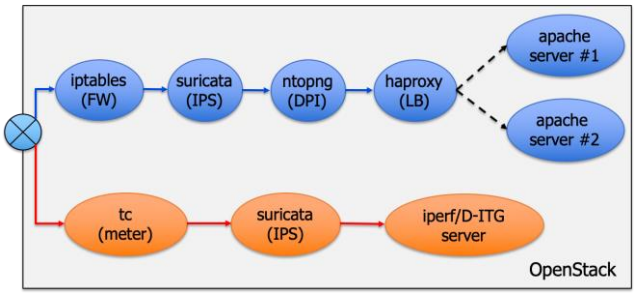


그림 3. 테스트베드 내 SFC 구성

V. 결론

본 논문에서는 인공지능 기반 NFV 관리 시스템의 구조를 제안하며 이를 실현하기 위한 테스트베드의 구현 내용을 설명한다. 향후 과제로 테스트베드 내 일부 오픈소스 플랫폼간 상호운영을 위한 추가 구현을 진행하며, VNF 및 SFC deployment 기능에 대한 기계학습 기반 구현 및 검증을 우선으로 제안된 인공지능 기반 NFV 관리 시스템을 확장해나갈 예정이다.

ACKNOWLEDGMENT

이 논문은 2019 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (2018-0-00749, 인공지능 기반 가상 네트워크 관리기술 개발 및 2015-0-00575, 글로벌 SDN/NFV 공개소프트웨어 핵심 모듈/기능 개발)

참 고 문 헌

[1] Tipantuña, Christian, and Paúl Yanchapaxi. "Network functions virtualization: An overview and open-source projects." 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM). IEEE, 2017.

[2] Boutaba, Raouf, et al. "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities." Journal of Internet Services and Applications 9.1 (2018): 16.

[3] Kourtis, Michail-Alexandros, et al. "T-NOVA: an open-source MANO stack for NFV infrastructures." IEEE Transactions on Network and Service Management 14.3 (2017): 586-602.

[4] OpenStack Foundation. “Taker - OpenStack NFV Orchestration.” https://wiki.openstack.org/wiki/Tacker.

[5] Mestres, Albert, et al. "Knowledge-defined networking." ACM SIGCOMM Computer Communication Review 47.3 (2017): 2-10.

[6] Han, Bo, et al. "Network function virtualization: Challenges and opportunities for innovations." IEEE Communications Magazine53.2 (2015): 90-97.

[7] Heegon Kim, et al. “A Machine Learning-based Method for Virtual Network Function Resource Demand Prediction.” 5th IEEE Conference on Network Softwarization (Netsoft 2019).

5G MEC 환경에서의 종단간 지연 시간 측정 기법 연구

현종환, 유재형*, 홍원기

포항공과대학교 컴퓨터공학과
*포항공과대학교 정보통신대학원

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

End-to-end Latency Measurement in 5G MEC Environment

Jonghwan Hyun, Jae-Hyoung Yoo*, James Won-Ki Hong

Department of Computer Science and Engineering, POSTECH
*Graduate School of Information Technology, POSTECH

요 약

5G 모바일 네트워크의 특징 중 하나는 1ms 미만의 지연 시간을 제공하는 것으로, 이러한 특징을 바탕으로 다양한 성능 요구조건을 가지는 서비스들이 가능해진다. 통신 사업자 관점에서는 이러한 요구조건을 만족시킬 수 있도록 각 서비스 별 지연 시간을 모니터링하는 것이 중요하다. 본 논문에서는 패킷 단위 텔레메트리 수집이 가능한 네트워크 모니터링 프레임워크인 INT(In-band Network Telemetry)를 활용하여, 5G MEC 환경에서 각 VNF 의 동작에 영향을 주지 않고 종단 간 지연 시간을 측정할 수 있는 방법을 제안한다. 본 논문에서 제안한 방법을 활용하면 실제 데이터 패킷이 전달될 때의 지연 시간을 정확히 측정할 수 있으며, 각 VNF 가 패킷을 처리하는 데 소요되는 시간도 측정할 수 있다.

Ⅰ. 서 론

5G 모바일 네트워크의 특징 중 하나는 고신뢰 초저지연 통신 (URLLC: Ultra-Reliable Low Latency Communication)으로, 1ms 미만의 지연 시간을 제공하는 것을 목표로 한다 [1]. 이를 가능하게 하는 주요 기술로는 MEC (Multi-access Edge Computing)[2]를 들 수 있다. MEC 는 통신 사업자의 코어망 외부에 위치한 클라우드에서 제공하던 각종 서비스들을 기지국 단위로 배치시키는 분산 클라우드 기술과 NFV (Network Function Virtualization)를 적용하여 지연 시간을 줄일 수 있는 기술이다.

5G 환경에서는 이러한 초저지연성을 바탕으로 현재의 4G 환경에서 불가능했던 여러 서비스들 (e.g., 자율 주행, 공장 자동화, AR/VR)을 가능케 한다. 이러한 서비스들은 서비스 특성에 따라 서로 다른 지연시간 및 전송 속도 등의 성능 요구조건을 가지며 [3], 이는 SLA (Service Level Agreement) 형태로 기술된다. 네트워크 사업자 관점에서는 각 서비스의 SLA 요구사항을 만족시키기 위해 각 서비스 별 지연 시간을 정확히 측정할 수 있어야 한다.

이러한 요구조건을 만족시키기 위해 INT (In-band Network Telemetry) [4]와 같은 네트워크 모니터링 프레임워크를 사용할 수 있다. INT 는 네트워크 텔레메트리 정보 (e.g., 패킷이 스위치를 통과할 때의 시간, 큐 상태 정보, 홉 간 지연, 스위치 ID)를 패킷 단위로 수집할 수 있는 기법으로, 패킷이 각 스위치를 지날 때마다 INT 헤더에 지정된 메타데이터를 패킷에 삽입하여 각 스위치 별 텔레메트리 정보를 수집한다.

INT 동작 방식은 그림 1과 같으며, INT 를 지원하는 스위치는 아래의 세 가지 모드로 동작한다.

- Source: INT Watchlist 에 매치되는 데이터 패킷에 INT 헤더를 삽입
- Transit: INT 헤더에 명시된 종류의 텔레메트리 정보를 패킷에 삽입
- Sink: 패킷에 삽입된 INT 헤더 및 텔레메트리 정보를 추출하여 텔레메트리 리포트 [5]를 생성 후 수집 장치로 전달하며, 원래의 데이터 패킷을 복원하여 호스트로 전달

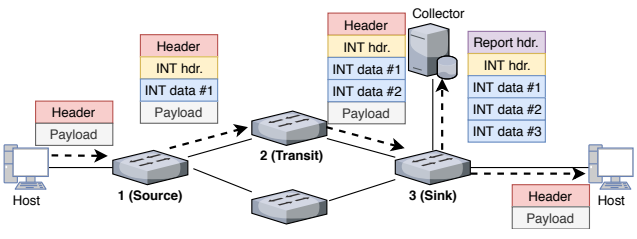


그림 1 INT 동작 방식

INT 를 활용하면 각 패킷의 종단 간 지연 시간을 정확히 측정할 수 있지만, MEC 환경은 일반적으로 가상화된 환경으로 구축되어 있다. 각 패킷은 미리 정의된 SFC (Service Function Chain)을 따라 여러 개의 VNF 를 거쳐 최종 목적지까지 전달된다. 이러한 환경에서 INT 를 사용하여 모니터링을 수행할 경우 문제가 발생할 수 있다. 각 VNF 로 패킷을 전달할 때

텔레메트리 정보를 추출하지 않고 전달하는 경우, 최종 목적지까지의 종단 간 지연 시간을 정확히 측정할 수 있지만, IPS (Intrusion Prevention System), DPI (Deep Packet Inspection) 등과 같이 패킷의 페이로드를 분석하는 VNF 에서는 INT 헤더 및 텔레메트리 정보를 페이로드로 인식하여 오동작이 발생할 수 있다. 이러한 현상을 방지하기 위해서는 각 VNF 에 연결된 스위치가 Sink 스위치로 동작하여 원래의 패킷으로 복원 후 전달하여야 하는데, 이 경우 하나의 패킷이 전달되는 과정에서 텔레메트리 정보가 VNF 개수만큼의 나누어지기 때문에 종단 간 지연 시간을 측정할 수 없다.

본 연구에서는 위와 같은 문제를 해결하여 MEC 환경에서도 종단 간 지연 시간을 정확하게 측정할 수 있는 방법을 제안한다.

Ⅱ. 본론

본 논문에서는 MEC 내부에서의 지연 시간 측정 방법만을 다루며, 제안하는 측정 방법은 그림 2와 같다.

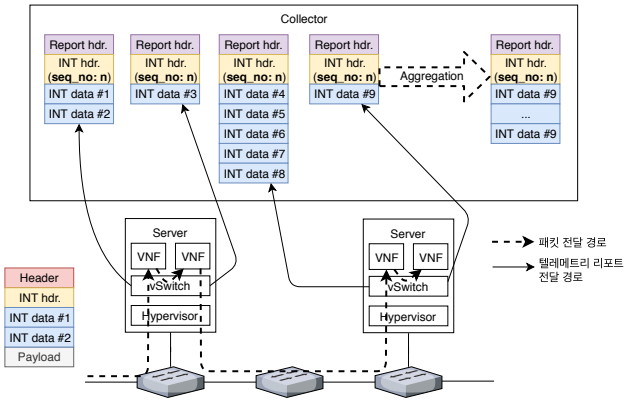


그림 2 MEC 환경에서의 INT 데이터 수집 구조

하나의 패킷이 여러 VNF 를 거쳐 최종 목적지까지 도달하는 과정에서 VNF 개수만큼의 추가적인 텔레메트리 리포트 패킷이 생성되어 수집 장치로 전달된다. 수집 장치에서는 하나의 패킷으로부터 생성된 리포트 패킷들을 조합함으로써 하나의 완성된 종단 간 텔레메트리 정보를 수집할 수 있다.

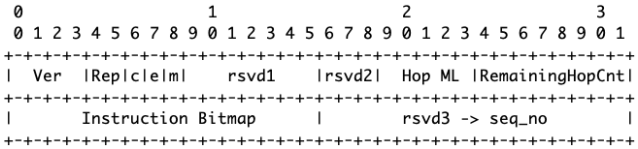


그림 3 제안하는 INT 헤더 구조

이를 위해서는 기존 INT 헤더[6]에 그림 3과 같이 Sequence number 필드를 추가하여야 한다. 제안한 지연 측정 기법에서는 하나의 패킷에 대해 여러 개의 부분적인 텔레메트리 리포트가 생성되기 때문에 이 리포트들을 조합하여 하나의 완성된 리포트를 만들기 위한 identifier 가 필요하다. 같은 5-tuple 과 sequence number 를 가진 리포트 패킷들을 조합하면 하나의 완성된 종단간 텔레메트리 정보를 수집할 수 있다.

본 연구에서는 INT 헤더의 사용되지 않는 필드(rsvd3)에 16-bit 의 sequence number 필드를 추가하고, INT Source 스위치에서 플로우 내의 각 패킷에 고유한 sequence number 를 추가해 준다. Sequence number 는 패킷이 VNF 를 통과할 때마다 보존되어야 하기 때문에 이러한 역할을 수행하는 스위치를 INT Anchor 스위치로 정의한다.

Anchor 스위치는 INT 헤더가 포함된 패킷을 VNF 로 전달하는 경우에는 Sink 모드로, VNF 에서 패킷을 수신하는 경우에는 Source 모드로 동작한다. Sink 모드에서는 INT Sink 기능을 수행함과 동시에, INT 헤더에서 보존하여야 할 필드 (e.g., Sequence number, Remaining Hop Count, M (MTU exceeded) bit)를 추출하여 스위치 레지스터에 저장한다. INT 헤더 및 INT Shim 헤더의 나머지 필드는 INT Source 에서 헤더를 추가하였을 때와 동일한 값을 가지므로 패킷이 VNF 로부터 Anchor 스위치로 전달되었을 때 INT Source 스위치로 동작하게 되면 헤더 추출 전과 동일한 값을 유지할 수 있지만, 위 세 가지 필드는 초기값과 다른 값을 가지게 되기 때문에 Anchor 스위치에서 보존해 주어야 한다. 이를 위해 새로 정의한 레지스터 목록은 표 1과 같다. 우선, Anchor Sink 에서는 추출한 INT 헤더 필드를 R_{Hdr} 에 저장하는데, 이 때 직전에 처리한 INT 패킷의 Sequence number (Seq_{n-1})를 R_{sink} 로부터 읽어온다. 그리고 R_{Sink} 와 $R_{Mapping}$ 에 각각 값을 업데이트해 준다. Anchor Source 에서는 INT Source 와 동일하게 INT 헤더를 추가하고 레지스터에 저장된 필드 값을 복원해 준다. R_{Source} 로부터 Seq_{n-1} 값을 가져온 후 $R_{Mapping}$ 으로부터 Seq_n 값을 읽어온다. 이후 R_{Hdr} 로부터 저장된 필드 값을 가져와 INT 헤더에 추가함으로써 INT 헤더 복원이 완료된다.

표 1 Anchor 스위치 레지스터 목록

Register	Key	Value
R_{Sink}	5-tuple, Egress port	Seq_{n-1}
R_{Source}	5-tuple, Ingress port	Seq_{n-1}
$R_{Mapping}$	5-tuple, Seq_{n-1}, (Ingress/Egress) port	Seq_n
R_{Hdr}	5-tuple, Seq_{n-1}, (Ingress/Egress) port	Seq_n, M bit, RemainingHopCnt,

위 과정을 통해 Anchor 스위치는 패킷의 INT Sequence number 를 보존함으로써 패킷이 VNF 를 통과하더라도 같은 Sequence number 를 가질 수 있도록 한다. 그리고 VNF 로 패킷을 전달할 때의 타임스탬프와 패킷을 다시 수신하였을 때의 타임스탬프도 텔레메트리 데이터로 수집함으로써 VNF 에서의 지연 시간도 측정할 수 있다.

Ⅲ. 결론

본 논문에서는 5G MEC 와 같은 VNF 환경에서 INT 를 활용하여 개별 패킷의 종단 간 지연 시간을 정확하게 측정하는 방법에 대해 제안하였다. 본 논문에서 제안한 방법을 활용하면 각 서비스 별 종단 간 지연 시간을 패킷 단위로 정확하게 측정할 수 있으며, 네트워크에서의 지연 시간과 VNF 에서의 지연 시간도 정확히 구분할 수 있다. 향후 연구로는 제안한 방법을 P4[7]를 사용하여 구현하며 실험을 통해 성능 및 정확도를 측정할 예정이다.

ACKNOWLEDGMENT

이 논문은 2017 년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No. 2017-0-00195, 멀티 서비스를 지원하는 프로그래머블 스위치 제어 기술 개발)

참 고 문 헌

[1] GSMA, “Road to 5G: Introduction and Migration”, April 2018.

[2] ETSI GS MEC 003 V1.1.1, “Mobile Edge Computing (MEC); Framework and Reference Architecture” , Mar. 2016.

[3] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, “A Survey on Low Latency Towards 5G : RAN , Core Network and Caching Solutions”, IEEE Communications Surveys & Tutorials, Vol. 20, No. 4, 2018, pp. 3098-3130.

[4] C. Kim, A. Sivaraman, N. Katta, A. Bas, A. Dixit, L. J. Wobker, and B. Networks, “In-band Network Telemetry via Programmable Dataplanes,” in Sosr, 2015, pp. 2- 3.

[5] The P4.org Applications Working Group . “Telemetry Report Format Specification v1.0” [online]; https://github.com/p4lang/p4-applications/blob/master/docs.

[6] The P4.org Applications Working Group . “In-band Network Telemetry (INT) specification v1.0”, [online]; https://github.com/p4lang/p4-applications/blob/master/docs.

[7] P. Bosshart, G. Varghese, D. Walker, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, and A. Vahdat, “P4: Programming Protocol-Independent Packet Processors,” ACM SIGCOMM CCR, vol. 44, no. 3, pp. 87- 95, 2014.

SDN 기반 QoS 보장형 Fast BSS Transition

황현동, 김영탁*
영남대학교 정보통신공학과

mch2d@ynu.ac.kr, ytkim@yu.ac.kr*

SDN Based QoS support Fast BSS Transition

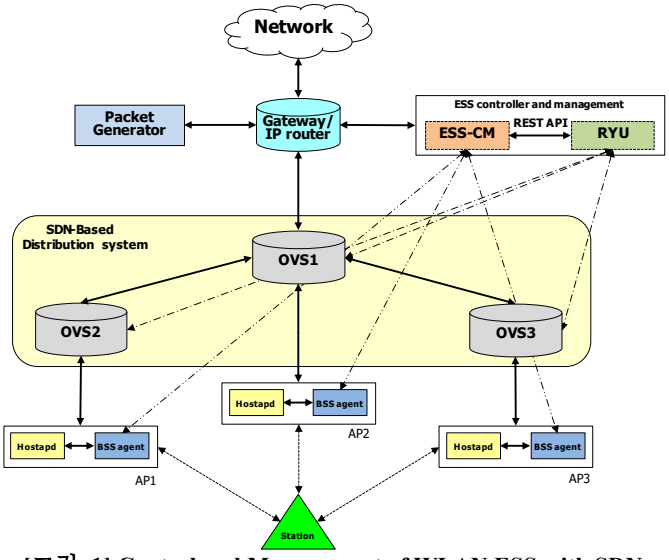
Hwang Hyundong, Kim young-tak*
Department of Information and Communication Engineering, Yeungnam University

요 약

무선 랜을 이용한 고품질의 실시간 모바일 멀티미디어 서비스를 제공하기 위해서는 QoS 를 보장한 BSS 이관 기법이 필요하다. 하지만 대부분 WiFi AP 들은 Fast BSS Transition 을 제공하지 않으므로 현재 연결된 AP 를 다른 AP 로 변경하는데 많은 시간이 소요되며, 모바일 서비스는 중단된다. 또한 특정 AP 에 집중된 트래픽을 분산시키기 위한 중앙 관제형 트래픽 엔지니어링도 요구된다. 본 논문에서는 끊김 없는 모바일 멀티미디어 서비스 및 로드 밸런싱을 위한 SDN 기반 QoS 보장형 Fast BSS Transition 기법을 제안한다. 제안된 기법은 후보 대상 AP 를 선택하기 전에 자원 가용성을 검사할 수 있는 SDN 기반 분산 시스템을 갖춘 WLAN ESS-CM 시스템과 QoS 를 보장한 Fast BSS Transition 기법을 제안한다. SDN 기반 분산 시스템은 단말기 이관 이전에 네트워크 경로를 변경과 패킷 버퍼링 기법을 이용해 실시간 데이터 서비스 제공에 QoS 를 보장해 주며, 실제 IEEE 802.11n 기반의 실험환경을 구축 및 제안 기법들을 구현하여 성능 측정하였다. 측정 결과 기존 핸드오버 방식과 비교하여 서비스 중단 시간 및 패킷 손실을 최소화하여 향상된 실시간 멀티미디어 서비스를 제공하였다.

Ⅰ. 서 론

IEEE 802.11 무선랜 기반의 실시간 모바일 멀티미디어 서비스를 제공하기 위해서는 BSS(Basic service sets)간 핸드오버 기능과 ESS(Extended service set)에서의 트래픽 부하 분산 기법이 필요하다. ESS 에서 AP(Access point)/BSS 들은 서비스 셀 범위와 전체 채널 이용률을 고려하지 않고 무분별하게 설치, 운용되어 지고 있으며 단말기는 AP 의 신호세기에 의해서만 접속되어 특정 AP 에게 부하가 발생한다. 이를 해결하기 위해서는 중앙관리 시스템에서 특정 AP 에 집중된 부하를 감지해 단말기들을 부하가 적은 AP 로 이관하는 기능이 필요하다. 실시간 멀티미디어 서비스를 제공하기 위한 핸드오버기법은 IEEE 802.11r 표준의 Fast BSS Transition(FT)기법이 있다. 하지만 IEEE 802.11r FT 는 핸드오버시 발생되는 서비스 단절시간 및 핸드오버 이후 백본 네트워크의 경로 변경시간에 발생되는 실시간 데이터의 패킷 지연 및 손실을 고려하고 있지 않고 있다.



[그림 1] Control and Management of WLAN ESS with SDN-based Distribution System

따라서 본 논문에서는 ESS-CM(Control & Management)을 이용해 AP/BSS 의 트래픽 정보를 수집, 부하가 발생한 AP 의 이관대상 단말기를 선정 및 실시간 데이터의 손실과 지연시간을 줄이는 SDN 기반 QoS 보장형 Fast BSS Transition 기법을 제안한다. [그림 1]은 전체 시스템의 구성도를 보여준다. QoS 보장형 Fast BSS Transition 은 핸드오버 이전에 백본 네트워크의 패킷 전송 경로를 SDN 컨트롤러를 이용해 변경하며, 대상 AP 로 미리 전송된 데이터를 저장 후 핸드오버 이후 전송함으로써 핸드오버시 발생하는 패킷 손실을 줄여 실시간 모바일 멀티미디어 서비스 품질을 보장할 수 있다. 논문의 구성은 다음과 같다. 2 장에서는 ESS-CM 을 이용한 무선랜 관리 시스템과 IEEE 802.11r Fast BSS Transition 을 설명하며, 3 장에서는 ESS-CM 과 SDN 컨트롤러를 이용한 네트워크 구성 및 ESS 내의 BSS 들의 트래픽 정보 수집방법에 대해 설명한다.

4 장에서 SDN 기반 QoS 보장형 Fast BSS Transition 에 대해 설명한다. 5 장에서는 IEEE 802.11r 기반의 Fast BSS transition 기법과 제안한 기법과의 성능 비교를 하며, 6 장에서 결론으로 마무리 짓는다.

Ⅱ. 관련연구

1. ESS 내의 AP/BSSs 들의 QoS 보장 로드 밸런싱

공공시설(공항, 터미널, 회의장), 캠퍼스에 설치된 WiFi 네트워크들은 분산 시스템에 연결되어 있으며 이를 통해 게이트웨이나 라우터에 접속되어 진다. 또한 인접 AP 들과는 직교 RF 채널을 사용함으로써 서로 간 간섭을 최소화하여 운용하고 있다. 따라서 모바일 단말기는 AP/BSS 간의 핸드오버를 할 시 채널을 변경하며, 해당 AP 에게 re-association 과정을 반드시 수행하여야 한다.

ESS 내의 AP/BSS 들의 총 처리량을 향상시키기 위해서는 각 AP 의 수용 용량을 계산하여 분배해 주는 기능이 필요 하다. 예를 들면 IEEE 802.11b WLAN 은 11Mbps 의 처리량을 수용 가능하지만 IEEE 802.11n/ac 는 300/600Mbps 의 처리량을 보인다. 따라서 수용가능한 단말기 숫자는 IEEE 802.11n/ac WLAN 이 더 많으므로, 더 넓은 채널범위로 운용되어져야 한다. 따라서 현재 WiFi 단말기들은 가장 가깝고 비컨 신호세기가 가장 강한 AP 에게만 접속되어 트래픽이 집중되어 네트워크 혼잡이 발생한다. 따라서 ESS 내의 총 트래픽을 고루 분배해주는 트래픽 엔지니어링이 필요하다.

2. IEEE 802.11r Fast BSS Transition

IEEE 802.11r fast BSS transition 은 단말기의 BSS 간 이관 시 발생하는 서비스 단절시간을 줄여주는 기법이다. Fast Transition(FT) 프로토콜은 ESS 도메인내의 동일한 SSID 를 가진 BSS 간 단말기 이관을 지원한다. BSS 이관에 서비스 단절시간을 최소화하기 위해 IEEE 802.11r 은 “Over-the-DS” Fast BSS Transition 기능을 제공하고 있다. 이를 위해 FT 는 4 가지 메시지(FT request, FT response, FT confirm, FT ack)를 제공하고 있으며, 대상 AP 와 현재 AP 간의 분산 시스템을 이용해 교환한다. 이러한 FT 메시지에는 RIC(resource information container)를 포함하고 있으며, 각 AP 의 트래픽 리소스 정보들을 사전에 교환하여, 이관 여부를 판단하게 된다.

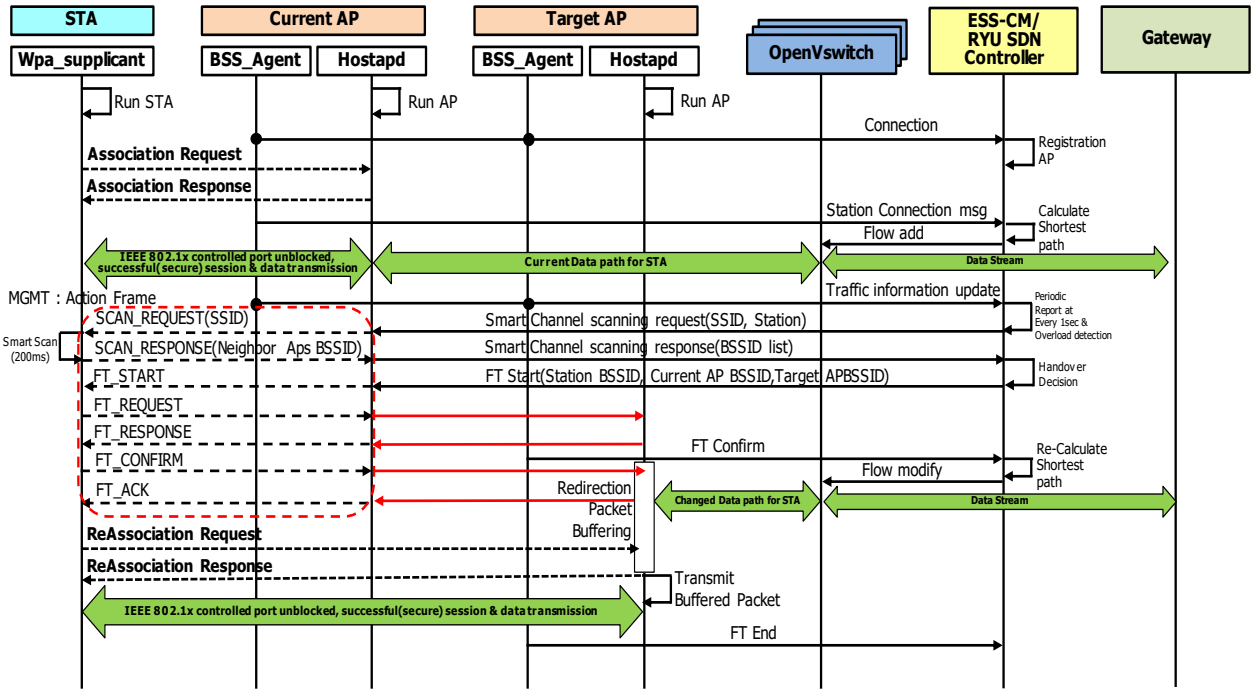
Ⅲ. QoS 보장형 무선랜 관리시스템 및 SDN 컨트롤러

1. ESS-CM 기반 QoS 보장형 무선랜 관리시스템

그림 1.은 ESS-CM 과 SDN 기반의 분산 시스템 구조이다. ESS-CM 은 하나의 ESS 의 BSS 들의 트래픽 정보와 단말기 접속 정보들을 관리한다. BSS 는 단말기와의 인증 및 접속 관리를 위한 Hostapd 와 ESS-CM 에게 트래픽 정보와 단말기 접속 정보를 제공하기 위한 BSS agent 로 구성된다. BSS agent 는 단말기 접속 시 ESS-CM 에게 BSSID 및 해당 단말기의 실시간 트래픽 정보를 1 초 간격으로 전송한다. ESS-CM 은 수집된 BSS 별 단말기 정보들과 전체 ESS 내의 트래픽 정보를 수집 및 핸드오버 여부를 판단한다.

2. RYU SDN 컨트롤러

본논문은 ESS 내의 분산 시스템을 OpenVswitch 와 RYU SDN 컨트롤러를 이용해 구성하였으며, RYU 컨트롤러는 ESS-CM 과 RESTful API 를 이용해 접속되어 있다. SDN 컨트롤러는 OpenVswitch 로 구성된 전체 네트워크의 구조 정보를 수집하며, 단말기 접속 및 이관 시 네트워크 경로를 OpenFlow 프로토콜을 이용해 OpenVswitch 의 Flow table 을 업데이트 한다. RYU 컨트롤러와 ESS-CM 은 네트워크 구조 정보를 서로 공유하며, 단말기 접속 시 백본 네트워크의 최단 거리를 계산하여 SDN 컨트롤러를 이용해 Flow table 을 업데이트 한다.



[그림 2] SDN 기반 QoS 보장형 Fast BSS Transition 시퀀스 다이어그램

Ⅳ. SDN 기반 QoS 보장형 Fast BSS Transition

1. ESS-CM 을 이용한 Fast BSS Transition

QoS 보장형 Fast BSS Transition 을 하기 위해서는 단말기는 주위 AP 리스트를 Smart Scan 기법으로 파악한 후 ESS-CM 에게 알려주며, ESS-CM 에서는 수집된 AP 들의 네트워크 자원 정보들과 단말기로부터 전달받은 AP 리스트를 이용하여 최적의 대상 AP 를 선정한다. 선정된 AP 로의 이관 명령은 ESS-CM 에서 FT Start 메시지를 현재 단말기가 접속된 AP 에게 전송한 후 AP 에서 다시 단말기로 Action frame 을 이용해 보내게 된다. 본 논문에서는 Action Frame 의 Fast BSS Transition 카테고리에 FT Start 와 FT End 메시지를 추가하였다. 이 두가지 메시지를 이용해 단말기에서 이관 결정을 하지 않고 중앙관리시스템에서 이관 결정과 명령 전달을 가능하게 하였다.

2. SDN 기반 백본 네트워크의 사전 경로 설정 및 패킷 버퍼링 기법

기존 Fast BSS Transition 기법은 AP 와 단말기간의 핸드오버 과정만 고려하고 있다. 따라서 단말기 까지의 Layer2 switch 의 Forwarding Table 은 IEEE 802.2 기반의 Logical Link Control(LLC) 프로토콜에 의해 단말기가 Association 이후 데이터 포트가 오픈이 되면 LLC Xid 패킷을 전송하게 되며, 이를 수신한 Switch 에서 Forwarding table 을 업데이트 한다. 핸드오버 이후 실시간 데이터 스트림들은 Forwarding Table 이 업데이트 되기 전까지 기존 AP 로 전송되어지며 기존 경로의 데이터 스트림은 패킷 손실로 처리된다. 본논문에서는 EES-CM 이 주관, 사전 패킷 경로를 SDN 기반의 네트워크를 이용하여 능동적으로 변경하여 패킷 손실을 줄일 수 있는 시스템을 제안한다. 이관 결정을 한 ESS-CM 은 단말기로 핸드오버 요청을 보내게 되며, 단말기는 Fast BSS Transition 과정을 수행한다. Fast BSS Transition 절차 중 Target AP 는 FT RESPONSE 메시지를 받은 후 BSS_Agent 를 이용해 ESS-CM 에게 FT confirm 메시지를 보낸다. 이후 Target AP 는 해당 단말기 목적 주소로 전달되어져 오는 패킷들에 대해 버퍼링을 실시한다. ESS-CM 은 단말기 핸드오버 과정을 인지 SDN 컨트롤러를 이용해 네트워크 경로 변경을 요청한다. 따라서 OpenVswitch 는 변경된 경로정보를 이용해 단말기가 이관되기 이전에 패킷들을 Target AP 로 전송할 수 있다. Target AP 는 버퍼링 패킷들을 단말기의 재 접속 절차를 완료 후 전송하여 실시간 데이터 손실을 대폭 줄일 수 있다.

3. SDN 기반 QoS 보장형 Fast BSS Transition 절차

[그림 2]는 본논문에서 제안한 SDN 기반 QoS 보장형 Fast BSS Transition 의 전체 절차이다. 초기 AP/BSS 들을 설치 시 ESS-CM 에게 BSS_Agent 들은 접속을 하게 되며, ESS-CM 은 BSS 들의 BSSID, SSID, 정보들을 수집하여 테이블로 가지고 있는다. 또한 SDN 컨트롤러를 이용해 네트워크 구조 정보를 수집하여 가지고 있는다.

Ⅴ. 성능분석

1. SDN 기반 QoS 보장형 Fast BSS 성능 측정 시나리오

[그림 1]과 같이 3 개의 AP 에 각 OpenVswtich 로 구성된 Layer2 Switch 를 설치하였다. AP 들은 Linux 기반의 Hostapd 2.6 버전을 이용하여 구성하였으며, OpenVswtich 2.5.2 버전을 이용하여 구성하였다. 단말기는 AP1 에서 시작하여 AP2, AP3 순으로 핸드오버를 실시하였으며, 단말기는 Wpa_supplicant 2.6 버전을 이용하였다. 측정을 위한 트래픽 생성은 위치는 Gateway/router 외부에서 발생시켰으며, UDP 데이터 125byte 를 초당 1360 개(1.33Mbps)를 단말기에 전송하였다. 핸드오버시 발생하는 서비스 단절시간을 정밀하게 측정하기 위해서 패킷 간격을 0.73ms 으로 전송하였다.

2. SDN 기반 QoS 보장형 Fast BSS Transition 성능분석

[표 1]은 DS 와 핸드오버 기법에 따른 패킷 손실과 서비스 단절시간을 측정한 결과이다. SDN 기반 Fast BSS transition 과 패킷 버퍼링을 하였을 때 평균 서비스 단절 시간은 8.09ms 이며, 패킷 손실은 11 개가 발생하였다. 실험 결과는 실제 AP 와 단말기를 IEEE 802.11n 기반의 무선랜 환경에서 실시하였으며, 기존 방식에 비해 이관 시 발생하는 서비스 단절시간과 패킷 손실률을 줄임으로써 실시간 데이터에 대한 QoS 를 보장한 로드밸런싱을 가능하게 한다. 또한 SDN 기반의 DS 을 구성함으로써 모바일 무선 네트워크 환경에서의 가변적인 네트워크 경로를 능동적으로 변경함으로써 실시간 데이터에 대한 QoS 를 보장할 수 있다.

[표 1] 패킷 손실 및 서비스 단절시간

QoS performance	Packet loss	Service Disruption time[ms]	Packet loss	Service Disruption time[ms]	Packet loss	Service Disruption time[ms]	Packet loss	Service Disruption time[ms]
Distribution system	Linux switch				SDN			
Handover method	without 802.11r		Fast BSS transtion					
Packet buffering	No						Yes	
Min	42	30.88	19	13.97	18	13.24	9	6.62
Avg	66	48.53	28	20.59	23	16.91	11	8.09
Max	122	89.71	47	34.56	27	19.85	14	10.29

Ⅵ. 결론

본논문에서는 모바일 무선 네트워크 환경에서의 실시간 데이터의 서비스 단절시간을 최소화한 SDN 기반 QoS 보장형 Fast BSS Transition 기법을 제안하였다. 제안된 기법은 3 가지 단계로 나뉘게 된다. 첫번째 ESS 내의 BSS 들로부터 단말기별 트래픽 정보를 주기적으로 수집 후 혼잡이 발생한 BSS 로부터 단말기를 선정하여 Smart scan 명령을 전달한다. 선정된 단말기는 스캔을 통해 주변 BSS 를 빠르게 탐색 후 ESS 에게 BSS 리스트를 전달하며 전달된 BSS 중 선택하여 이관명령을 내린다. 두번째 이관 명령을 받은 단말기는 Fast BSS Transition 절차를 수행하게 되며, ESS-CM 은 SDN 컨트롤러를 이용해 대상 BSS 까지의 최적 경로를 계산해 OpenvSwtich 의 Flow table 을 업데이트 한다. 세번째 사전에 경로가 변경되어 대상 AP 로 전달된

데이터 스트림은 버퍼링 후 단말기 재접속시 전송하게 된다. 제안된 기법은 Ubuntu 14.04(Linux 3.16.0 kernel) 기반의 ath9k IEEE 802.11n 무선랜 드라이버에 구현하였으며, Hostapd 2.6 버전과 wap_supplicant 2.6 버전을 이용하여 무선랜 환경을 구성 및 구현하였다. 또한 Ryu 4.14 SDN 컨트롤러와 OpenVswitch 2.5.2 를 이용하여 DS 를 구성하며, ESS-CM 과의 연동기능을 구현하였다.

실험 결과에 따르면 SDN 기반 Fast BSS Transition 기법은 기존 기법과 비교화여 서비스 단절 시간을 대폭 줄여주며, AP 들의 부하정보를 실시간 수집하여 필요한 네트워크 자원을 적절히 분배해 ESS 내의 총 처리량을 향상시킬 수 있다.

ACKNOWLEDGMENT

"This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2019-2016-0-00313) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation)"

참 고 문 헌

[1] IEEE Standard 802.11-2007, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification,” June 2007.

[2] IEEE Std 802.11r-2008, “Amendment 2: Fast Basic Service Set (BSS) Transition,” IEEE, July 2008.

[3] Hyundong Hwang, Young-Tak Kim, “Enhanced Fast BSS Transition on Enterprise WLAN with SDN-based Distribution System”, 13th CNSM, Nov. 2017

[4] Hyundong Hwang and Young-Tak Kim, “QoS-aware Fast BSS Transitions for Seamless Mobile Broadband Multimedia Service Provisioning and Load Balancing,” in Proc. Of International Conference on Consumer Electronics (ICCE) 2014, Las Vegas, USA.

[5] Ryu SDN controller, http://ryu.readthedocs.io/en/latest/.

[6] Open vSwitch 2.5.90 Documentation (Open vSwitch Manuals), http://openvswitch.org/support/dist-docs/.

IEEE 802.11ah/ Sub-1GHz 기반의 사물인터넷 스마트 제어

김민철, 김영탁*
영남대학교 기계 IT 대학 정보통신공학과

kmc724@ynu.ac.kr, *ytkim@yu.ac.kr

Smart Control in IEEE 802.11ah/Sub-1GHz for IoT Wireless Networking

Min-Cheol Kim, Young-Tak Kim*
Dept. of Information & Communication, Yeungnam University.

요 약

사물인터넷에서는 반경 1 Km 이내 구간의 다양한 위치에 최대 8000 개 정도의 단말장치가 설치될 수 있어 송신장치와 수신장치간의 전송 채널 상태에 따라 송신전력과 전송 속도를 조절할 수 있어야 하며, 주파수 간섭 파악하여 송신 채널 주파수를 변경할 수 있어야 한다. 본 논문에서는 IEEE 802.11ah/Sub-1GHz 기반의 사물인터넷 무선 네트워킹에서의 스마트 제어 기법을 제안하며, 제안된 기법의 성능을 측정하고 분석한다.

Ⅰ. 서론

사물인터넷 통신에서는 반경 1Km 정도의 영역에서 최대 8000개 정도의 사물인터넷 단말장치들이 다양한 위치에 설치될 수 있으므로 송신 장치와 수신 장치간의 무선 전송 채널 들이 다양한 상태에 있을 수 있으므로 AP에서는 매우 효율적인 채널 제어 기능이 제공되어야 한다[1-3]. 사물인터넷 단말장치는 제한된 배터리 용량으로 구동되기 때문에 에너지 소모를 최소화할 수 있어야 하며 전송채널의 상태에 따라 송신전력과 전송속도를 조절할 수 있어야 한다. 아울러 넓은 지역에 걸쳐 단말장치들이 분산되어 있어 다양한 주파수 간섭이 발생할 수 있어 AP는 주파수 간섭 발생을 신속하게 파악하고 채널을 변경할 수 있어야 한다. 특히, 사물인터넷 통신망에서 사용하는 900MHz 대역의 IEEE 802.11ah (WiFi-HaLow)에서는 데이터 전송 속도가 50Kbps ~ 4Mbps으로 제한되므로, 이를 효율적으로 사용할 수 있는 MAC 프로토콜이 적용되어야 한다.

본 논문에서는 Sub-1GHz 무선통신 채널을 보다 효율적으로 사용할 수 있는 CSMA/CA와 TDMA를 혼합하여 하이브리드 방식으로 사용하는 MAC 프로토콜을 기반으로 전송 채널 상태에 따라 송신전력과 전송 속도를 조절할 수 있으며 주파수 간섭 파악하여 송신 채널 주파수를 변경할 수 있는 스마트 제어 기법을 제안한다. 제안된 방식은 CC1312를 기반으로 구현하였으며, 실제 반경 1Km 영역에서 전송 성능을 측정하고 분석하였다.

Ⅱ. 관련 연구

사물인터넷 통신을 위하여 Wi-Fi Alliance가 Sub-1GHz 대역의 IEEE 802.11ah(Wi-Fi HaLow)에서는 CSMA/CA를 통한 충돌회피 방식을 표준화하고 있으며[4], 대부분의 국가들에서 900MHz 주파수 대역을 할당하고 있다. 한국에서는 무선설비규칙에서 RFID/USN 등의 무선 설비를 위하여 917 ~ 923.5MHz 주파수 대역에 200 KHz 대역폭의 32개 채널을 사용하도록 규정하고 있다. 이 주파수 대역에서의 채널 최대 송신 전력은 채널마다 정리하고 있으며, 채널 2, 5, 8, 11, 14 및 17에서는 최대 4W 이하, 채널20 ~ 32에서는 200mW이하로 규정하고 있다[5].

IEEE 802.11ah에서 표준화하고 있는 CSMA/CA 방식의 채널 접속 제어 방식은 8000개 규모로 많은 단말장치가 접속되는 중앙집중식 대규모 M2M환경에서는 성능이 저하되는 것으로 분석되어 CSMA/CA와 TDMA 방식을 혼용하는 Hybrid CSMA/CA-TDMA 접속제어 방식이 제안되고 있다[6].

CSMA/CA-TDMA 하이브리드 전송제어에서 AP는 C-slot과 T-slot으로 구성된 Beacon Period를 매 100ms 주기로 구성한다. 각 IoT 단말장치는 IEEE 802.11ah와 같이 AP의 Beacon Frame를 기준으로 Authentication/Association 절차를 거쳐 AP에 접속된다. AP는 Beacon Frame을 송신하여 모든 단말장치에 Super Frame의 시작을 알린다. IoT 단말장치는 C-slot에 Authentication Request (Auth_Req)를 AP에 전송한다. AP는 C-slot 시간 동안 수신하다가 C-slot 종료 후 각 IoT 단말장치들에 대한 응답을 Authentication Response (Auth_Resp)에 담아 한번에 보낸다. 이때 Authentication이 완료된 IoT 단말장치들에 대한 Time Slot 정보도 함께 포함되는데 이 정보는 각각 해당 IoT 단말장치의 Time Slot 크기, 개수, 위치이다. Time Slot의 위치는 각 IoT 단말장치가 Beacon Frame을 수신하는 지점부터 Mini-slot(0.5ms)으로 환산되어 전달된다. Auth_Resp를 받은 단말장치는 이후 T-slot에 진입하여 AP로부터 할당 받은 Time Slot을 이용해 Association 절차를 끝마친다.

Ⅲ. IEEE 802.11ah/Sub-1GHz 사물인터넷 무선 네트워킹 스마트 제어

3.1 IEEE 802.11ah/Sub-1GHz 무선 네트워킹 스마트 제어 구조

본 논문에서 제안하는 IEEE 802.11ah /Sub-1GHz무선 통신 구조는 하나의 중앙 집중형AP(Access)에 다수의 노드 들이 접속하는 star형 토폴로지를 구성하며, 등록 절차를 거쳐 모든 노드의 주소 정보를 파악하고, 사물인터넷 네트워크에서 사용할 수 있는 식별번호를 부여하여 전송 프레임의 주소로 사용한다. 100ms 간격의 Beacon Interval (BI)은 제어 메시지 전송을 위한 Control Message Exchange (CME) 구간과 데이터 전송을 위한Data Exchange (DE) 구간으로 구분된다. CME 구간은 안정적인 무선 통신 채널 상태를 보장받기 위해 최소 전송속도 및 최대 송신 전력으로 설정한다. 각 BI는 AP가 beacon프레임을 전송하면서 시작되며, beacon 프레임에는 그 BI 내에서 CSMA/CA 구간 할당 정보를 포함하여 각 IoT 단말 장치들이 언제 할당된

TDMA 슬롯을 사용할 수 있는지 통보한다.

사물인터넷 단말장치 들이 데이터를 전송하기 위해서는 사전에 CSMA/CA 구간 내에서 경쟁을 통해 Data 전송을 위한 TDMA Slot을 요청한다. 이때, 가변 할 전송속도 및 송신전력 정보를 전달하고, TDMA 구간에서 적용 하도록 한다. AP는 RA (resource allocation) 구간에서 전송 속도 및 송신 전력 변경을 최소화 하기 위한 scheduling을 통해 노드 들의 TDMA Slot을 할당하고 할당된 정보를 전송한다. 노드 들은 자신이 할당 많은 TDMA Slot에서 요청한 전송속도 및 송신전력으로 설정하고, Data Frame을 전송한다.

AP는 수신 받은 데이터 프레임들의 CRC 정보를 노드들에게 Ack frame을 통해 전송한다. 노드는 Ack frame의 CRC 및 PER정보를 기반으로 최대의 전송속도를 설정하고, 최대의 전송속도 링크 상태를 유지할 수 있는 최소의 송신전력 값을 찾아내어 사용할 수 있게 함으로써 에너지 효율적인 사물인터넷 통신을 가능하게 한다.

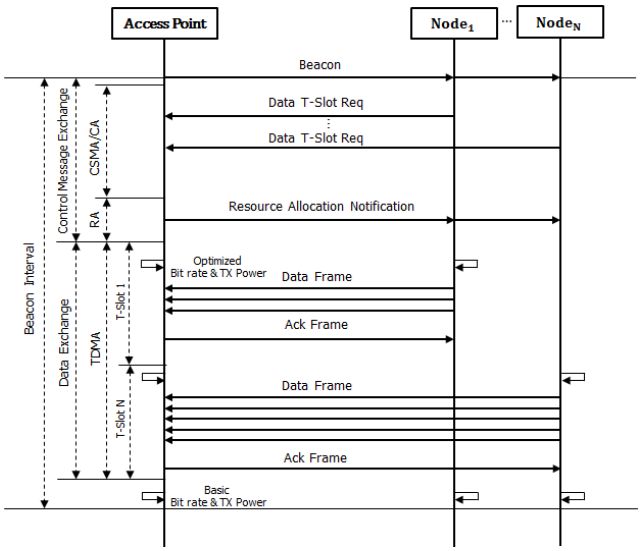


그림 1. Sub 1GHz 채널의 Hybrid CSMA/CA-TDMA MAC Sequence

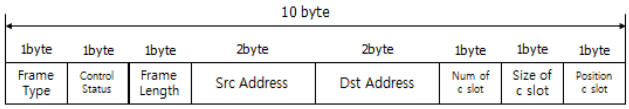


그림 2. Beacon Frame 구조

3.2 전송 채널 상태에 따른 송신 전력 제어

등록이 완료된 IoT 단말장치는Association Number를 이용해 AP와 통신할 수 있게 된다. 데이터를 전송하고자 하는 단말장치는 C-slot을 통해 T-slot Request를 전송하고 등록 단계와 마찬가지로 AP는 단말장치가 요청한 전송데이터의 크기, Frame개수를 보고 T-slot에서 이용하게 될 Time Slot의 크기, 개수, Time Slot의 위치 정보를 할당한 다음 하나의 T-slot Allocation에 담아 보낸다. 응답을 받은 단말장치는 다음 이어지는 T-slot 구간에 지정된 Time Slot을 이용하여 데이터를 전송하고 예정된 데이터가 모두 전송되면 AP는 Frame단위 비트 에러 발생 정보를 포함한 그림 3과 같은 Data Ack를 보낸다.

수신장치는 상대방 IoT 단말장치로부터 수신된 프레임의 수신 전력 (RSSI)와 프레임에 포함된 비트 에러 발생 여부에 따라 송신 전력을 조절할 수 있으며, 비트 에러가 발생되지 않는 최저 송신 전력과 최대 전송 속도로 변경한다.

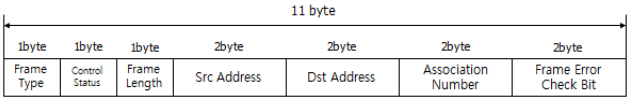


그림 3. Data Ack Frame 구조

3.3 채널 간섭 상태 파악 및 채널 변경

사물인터넷은 반경 1Km 범위 내에서 다양한 위치에 단말장치들이 설치되므로 송수신단간의 채널에는 다양한 주변 환경에 의한 주파수 간섭이 발생하게 된다. 수신단에서는 이러한 채널 간섭 상태를 신속하게 파악하여야 하며, 필요에 따라 채널을 변경할 수 있어야 한다.

채널 간섭 상태를 파악하는 방법으로는 수신 전력(RSSI)와 비트 에러 발생율을 기반으로 파악할 수 있는데 이를 위하여 채널 간섭이 없는 경우 (즉, 프레임의 비트 에러가 발생하지 않는 상황)에서의 전송 속도별 최저 수신 전력을 지속적으로 측정하여 기록하여야 한다.

이 후 프레임의 수신 전력이 이 최저 수신 전력 보다 일정 수준 이상으로 높은 경우에도 프레임의 비트 에러가 발생한다면 이는 송수신 채널상에 간섭이 발생된 것으로 판단할 수 있다.

IV. 구현 및 실험결과

4.1 IEEE 802.11ah/Sub-1GHz 통신 모듈 구현

본 논문에서 사용된 시험용 IEEE 802.11ah/Sub-1GHz 통신 모듈은 TI(Texas Instruments)사의 Sub-1GHz 저전력 RF Microcontroller인 CC1312R LaunchPad를 기반으로 구현하였다. 아울러 CC1310과 Range Extender인 CC1190을 함께 사용해 PA(Power amplifier)와 LNA(Low noise amplifier)기능을 사용하여 보조 실험을 하도록 구성하였다. 그림 4는 본 논문에서 구현한 IEEE 802.11ah/Sub-1GHz 모듈의 기능 블록도를 보여준다. TI사의 CC1312R은 625bps(Long Range Mode) ~ 4Mbps(High Speed Mode)까지 다양한 전송속도를 지원한다.

그림 4의 구조에서 CC1312R 모듈은 900MHz 무선 채널을 사용한 프레임 송신 및 수신 기능을 담당하며, Listen-before-Talk 기능을 구성할 수 있다. 실제 사물인터넷 응용 프로그램 기능은 Raspberry Pi 3 모듈에 설치된 Ubuntu 임베디드 Linux 운영체제와 CC1312R용 디바이스 드라이버 모듈을 사용하여 구현되어 있다.

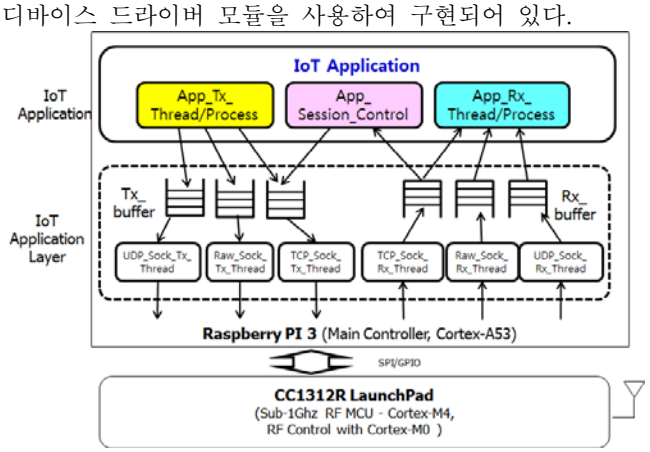


그림 4. 1 IEEE 802.11ah/Sub-1GHz 통신 모듈의 기능블록도

4.2 IEEE 802.11ah/Sub-1GHz 통신 모듈의 데이터 프레임 전달 성능 분석

IEEE 802.11ah/Sub-1GHz 통신 모듈의 에너지 효율적인 전송 기능을 갖추기 위하여 전송 거리, 전송 속도

별 비트에러 발생률을 측정하고 분석하였으며, 프레임에 비트에러가 발생하지 않는 범위에서 송신 전력을 최소한으로 사용할 수 있도록 하였다.

먼저 안정화된 전송속도인 50Kbps ~ 500Kbps 전송속도에서 실제 데이터 전송속도를 측정하였다. 그림 5는 AP가 노드에게 할당하는 TDMA Slot 크기를 고정하고, 전송속도를 가변 함에 따라서 증가하는 throughput을 측정하고 분석한 결과이다.

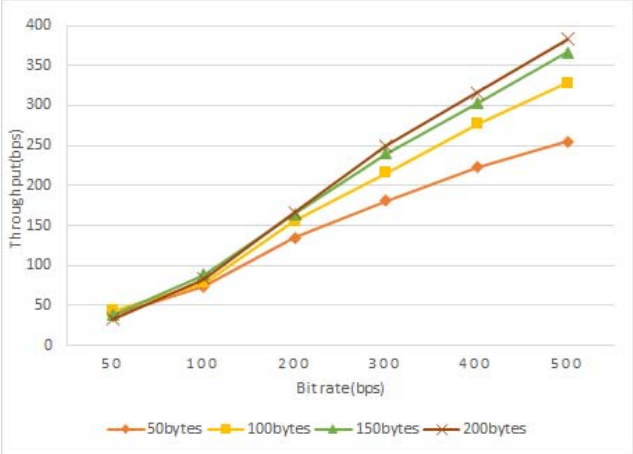


그림 5. Frame 크기 별 Throughput

CC1190은 High Gain Mode로 동작 시켜 CC1310 TX Power 파라미터 값을 설정하여 송신 전력을 12dBm ~ 26dBm로 변경할 수 있다. 그림6은 전송속도를 100Kbps, AP와 사물인터넷 단말장치 사이의 거리를 고정시켜 놓은 상태에서 TX Power에 따른 PER(Packet Error Rate)를 측정했다.

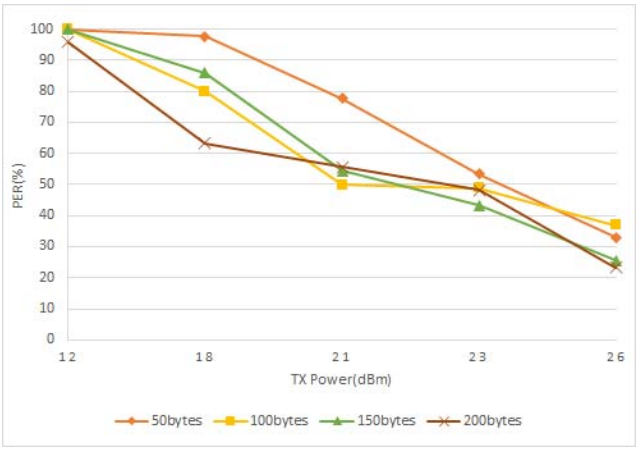


그림 6. 송신 전력과 프레임 에러 발생률의 추세 분석

4.3 전송 거리별 IoT 데이터 프레임 전달 성능 분석

900MHz 무선 전송 채널을 사용하여 반경 1Km 이내에 설치된 단말장치들을 연결하는 사물 인터넷에서 AP와 IoT 단말기 간의 거리가 멀어짐에 따라 전송 신호 전력이 감쇠되고 따라서 프레임 에러 발생률이 증가하게 된다. 전송거리와 함께 전송 신호 전력 감쇠에 영향을 주는 것이 전송 경로상에 건물이나 나무들이 있는 경우이며, 이러한 경우 추가적인 감쇠가 발생하여 송신단에서의 송신 전력을 증가시키거나 전송 속도를 낮추어야 한다.

그림 7는 전송거리와 전송 속도 별 프레임/패킷 에러 발생률을 분석한 그래프를 보여준다. 이 그래프에서 확인할 수 있는 것과 같이 전송 거리가 멀어짐에 따라 신호 감쇠가 발생하여 패킷/프레임 에러율이 증가하는 추세를 나타내며, 동일한 거리에서도 전송 속도가 높을수록 더 큰 프레임 에러가 발생하는 것을 알 수 있다. 따라서 AP와 IoT 단말장치간에 수신 전력과 에러 발생률을 지속적으로 측정하여 최적의 전송속도와 송신 전력을 사용할 수 있는 구성을 파악하여야 한다.

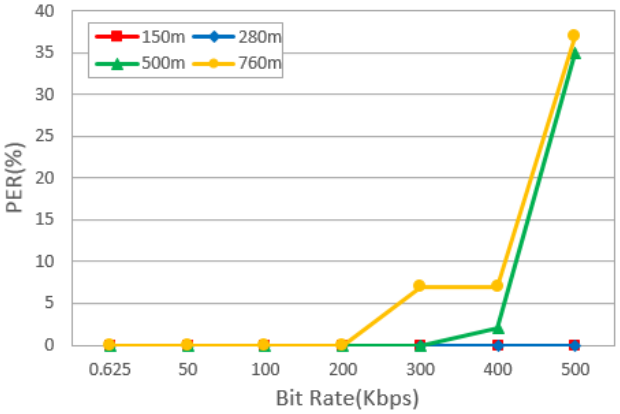


그림 7. 전송거리와 전송 속도 별 패킷/프레임 에러 발생률

4.4 채널 간섭 파악 및 채널 변경

사물인터넷에서는 무선 전송 신호의 전달 거리가 1Km 정도로 멀기 때문에 송신단과 수신단의 환경이 크게 다를 수 있다. 또한 무선 채널의 특성에 따라 채널 별로 수신 신호 전력이 다른 특성을 보일 수 있으며, 채널별 프레임 에러율로 다르게 나타날 수 있다. 표 1은 채널별 수신 신호 전력 (RSSI)값이 다르게 나타나는 것을 보여주며, 표 2는 채널별로 프레임 에러율이 다르게 측정되는 것을 보여 준다.

특히 수신단 주변에서 동일한 채널 주파수를 사용하는 다른 무선 통신 신호가 존재하면 이는 해당 채널을 사용하여 프레임을 전송하는 IoT 통신에 채널 간섭 (channel interference)로 작용하여 수신 신호 전력 (RSSI)가 강한 경우에도 주파수 간섭 잡음 때문에 프레임 에러가 크게 발생할 수 있다.

넓은 지역에서 900MHz 채널 통신이 이루어 져야하는 사물 인터넷 환경에서 주파수 채널별로 발생하는 채널 간섭 (channel interference)을 회피하기 위해서는 지속적으로 채널의 RSSI값과 프레임 에러율 (FER)을 측정하고 분석하여 주변 환경으로부터 채널 간섭이 발생하는 가를 확인하여야 한다.

만약 채널 간섭이 발생되는 것으로 확인되면 AP와 IoT 단말간에 사용하는 채널을 다른 채널로 변경하여 통신을 재 구성할 수 있어야 한다.

표 1. 채널 별 수신 신호 전력 (RSSI) (단위: dBm)

Tx Rate (BPS)	917MHz	918MHz	919MHz	920MHz	921MHz	922MHz	923MHz
50	-67.2	-67.6	-66.8	-68	-69.8	-71.6	-73
100	-67.4	-67.2	-68.2	-70.6	-72.4	-74	-75.6
200	-69.2	-69.2	-70.6	-72.6	-74.6	-78.2	-79.4
300	-68	-69	-70.2	-72	-75.4	-76.6	-79.2
400	-68	-68.6	-69.6	-70.8	-74.2	-77	-78.2
500	-67.4	-67.2	-68	-70	-69.4	-67.4	-67.6

표 2. 채널별 프레임 에러율

Tx Rate (BPS)	917MHz	918MHz	919MHz	920MHz	921MHz	922MHz	923MHz
50	0.00%	0.00%	0.00%	0.09%	0.02%	0.02%	0.32%
100	0.10%	0.00%	0.01%	4.82%	2.97%	1.24%	4.35%
200	0.40%	0.61%	0.21%	0.45%	0.64%	8.28%	27.50%
300	0.77%	0.77%	6.49%	0.83%	5.28%	35.10%	65.83%
400	1.82%	2.57%	10.23%	5.11%	38.12%	92.38%	87.60%
500	9.55%	0.62%	0.65%	0.68%	0.53%	0.87%	0.64%

V. 결론

본 논문에서는IEEE 802.11ah/Sub-1GHz 기반의 사물인터넷 통신에서 사용되는 무선통신 채널을 보다 효율적으로 사용할 수 있도록 CSMA/CA와 TDMA를 혼합하여 하이브리드 방식으로 사용하는 MAC 프로토콜을 기반으로 전송 채널의 스마트 제어 기법을 제안하였다. 제안된 방식에서는 CSMA/CA를 사용하여 등록 및 데이터 전송을 위한 채널할당을 요청하고, 할당된 TDMA slot을 사용하여 데이터를 전송함으로써, 대규모 사물인터넷 단말장치들이 경쟁하는 환경에서도 채널 이용율을 높일 수 있게 하였다. 제안된 기법은 CC1312R LaunchPad를 기반으로 모듈을 구현하였으며, 반경 1Km 구간에서 실제 전송능력을 측정하고 성능을 분석하였다.

ACKNOWLEDGMENT

"본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음" (IITP-2019-2016-0-00313).

참 고 문 헌

[1] Jie Lin et.el, " A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications", IEEE Internet of Things JOURNAL Vol. 3 NO.5, oct, 2017, pp. 1125-1142.

[2] Corrales Madueno, Cedomir Stefanovic and Popovski, " Reliable and Efficient Access for Alarm-Initiated and Regular M2M Traffic in IEEE 802.11ah Systems", IEEE INTERNET OF THINGS JOURNAL VOL. 3 NO.5, oct, 2016, pp. 673-682.

[3] Stefan Aust, Venkatesha Prasad and Ignas G. M. M. Niemegeers, "Outdoor Long-Range WLANs: A Lesson for IEEE 802.11ah," IEEE COMMUNICATION SURVEYS & TUTORIALS, vol. 17, no. 3, 2015.

[4] T. Adame, A. Bel, B. Bellalta, J. Barcelo, and M. Oliver, " IEEE 802.11AH: The WiFi approach for M2M communications," IEEE Wireless Communication, vol. 21, no. 6, pp. 144- 152, Dec. 2014.

[5] 무선 설비 규칙, 미래창조부 고시 2016-125 호, 2016. 11. 30.

[6] Nurullah Shahin, Rashid Ali and Young-Tak Kim., " Hybrid Slotted-CSMA/CA-TDMA for Enhanced Registration of Massive IoT Devices," IEEE Access Vol. 6, pp 18366 - 18382, 2018.

[7] 김기태, 김영탁, " 에너지 효율적인 사물 인터넷 통신을 위한 Sub-1GHz 무선 통신 채널의 전송 속도 및 송신 전력의 스마트 제어 기법", 한국통신학회 하계 학술대회, 2017.

[8] 정용환, 김영탁, " IEEE 802.11ah 기반의 사물인터넷 통신을 위한 CSMA/CA 와 TDMA 의 하이브리드 전송 제어 기법", 한국통신학회 하계 학술대회, 2017.

네트워크 보안을 위한 SIEM 솔루션 비교 분석

이종화, 방지원, 김종욱, 최미정
강원대학교
{a8478836, jiwonbang, goldbear564, mjchoi}@kangwon.ac.kr

Comparison of SIEM Solutions for Network Security

Jong-Hwa Lee, Jiwon Bang, Jong-Wouk Kim, Mi-Jung Choi
Dept. of Computer Science, Kangwon Univ.

요 약

기술이 발전함에 따라 사용자에게 가해지는 네트워크상의 최신 보안 위협이 늘어나고 있다. 해커가 악의적인 목적을 가지고 산업 또는 기업의 시스템을 공격함으로써 기밀정보가 유출되거나 사이버 테러, 정보 자산의 침해, 금전적인 손해 등의 많은 사회 문제를 야기한다. 복잡하고 다양해지는 위협으로 인해 현 보안 인력만으로 모든 위협을 탐지하고 분석하기에는 역부족인 상황이 되었다. 특히, 365일 24시간으로 돌아가는 산업 기반 시설에서 사용하는 SCADA(Supervisory Control And Data Acquisition)는 정적인 데이터를 수집 및 분석하므로, 실시간으로 발생하는 보안 위협에 대해서는 매우 취약하다. 본 논문에서는 실시간으로 시스템의 상태를 모니터링이 가능하고 보안 위협을 탐지하는 강력한 통합 보안 관리 시스템인 SIEM(Security Information and Event Management)에 대해 소개한다. 다음으로 다양한 기업의 SIEM 제품들과 오픈 소스로 배포되는 AlienVault 사의 OSSIM(Open Source SIEM) 을 비교분석하고, OSSIM을 이용한 활용 사례와 OSSIM을 활용할 수 있는 방안을 제시한다.

Ⅰ. 서 론

IT 기술이 발전하면서 사물 인터넷(Internet of Things, IoT), 증강 현실(Augmented Reality), 블록체인 등 제 4차 산업혁명이 일어나고 있다[1]. 보안 위협도 점차 고도화되면서 취약점 공격, 랜섬웨어, APT(Advanced Persistent Threat) 공격 등이 빠르게 확산되고, 끊임없이 새로운 위협이 등장하고 있다. 또한, 부족한 보안 인력과 자동화된 위협들에 비해 방어 및 탐지는 자동화가 쉽지 않다. 이러한 이유로 기업 및 산업 기반 시설에서는 자동화된 보안 관리 솔루션을 도입한다. 하지만 도입 기준이 불분명하여 적절한 제품을 찾지 못하고 있다.

산업 기반 시설 공정 및 설비 공정에서 사용하는 산업 자동화 및 제어 시스템(Industrial Automated Control Systems, IACS)의 주요 유형 중 하나는 감시 제어 및 데이터수집 시스템이다. SCADA는 본래 폐쇄적인 형태로 설계되었으나 기술이 표준화되고, 공개되면서 SCADA, 인터넷, 사내 네트워크 간 연결이 증가하여 편의성이 증대되었다. 하지만 접목된 기술들로 인한 취약점이 드러났고, 산업 시설을 감시하고 파괴하는 악성 소프트웨어인 ‘스턱스넷(Stuxnet)’ 공격이 개발되었다. 스턱스넷은 SCADA 시스템의 외부에서 SCADA 마스터 제어 스테이션 접속을 시도하고, RTU(Remote Terminal Unit)와 Local PLC(Programmable Logic Controller)를 분석하여 제어 시스템의 권한을 획득한 후 시스템을 공격한다. 뿐만 아니라 내부 위협으로 관리자의 실수로 인해 안전장치가 해제되고 기밀 정보가 누설 되는 등의 다양한 위험도 존재한다. SCADA 시스템은 특성상 사회 중요 기반 시설로서 작은 문제가 발생하더라도, 타 시스템과 비교할 수 없을 만한 피해가 나타난다. 따라서 SCADA 시스템은 내부와 외부의 모든 이벤트를 자동적으로 감시하고 로그를 분석하여 위험을 예방, 경고하는 강력한 보안 시스템이 필요하다[2].

본 논문에서는 SIEM의 대략적인 설명을 관련 연구에서 설명하고, 2018년 가트너 사에서 선정한 상위 SIEM 제품들과 오픈 소스로 배포되는 SIEM 제품인 AlienVault 사의 OSSIM에 관한 특징과 차이점을 3장에서 비교 분석한 후 4장에서 OSSIM을 활용한 사례와 활용 방안을 제시한다[3].

Ⅱ. 관련 연구

SIEM은 기존 SEM(Security Event Management)과 SIM(Security Information Management)이 결합된 솔루션이며 다량의 로그 데이터와 이벤트들을 실시간으로 상관분석을 수행하는 시스템을 의미한다. 기존의 보안 기술들은 방화벽, 안티 바이러스와 같이 주로 하나의 공격 위협 요소에 대응하지만, SIEM은 각종 보안 및 네트워크 장비로부터 수집되는 다양한 정보를 이용한다. 사용자는 클라이언트를 통해 해당 서비스를 제공하는 SaaS(Service as a Service) 또는 클라우드 방식이 도입되어 하이브리드 IT 인프라가 사용되고 있다. 뿐만 아니라 IT 기술에 발전에 따라 빅 데이터 기술, 기계 학습, AI 등의 최신 IT 기술들과의 융합이 진행되고 있다[4, 5, 6].

SIEM은 보안 데이터를 기록하고 규정 준수를 위한 보고서를 생성한다. 네트워크의 경계부터 최종 사용자까지 전체 범위에 있는 네트워크 장비 및 보안 장비들의 로그를 수집, 저장 및 분석을 할 수 있는 효율적인 통합 로그 관리 기능과 사고 대응 및 포렌식, 알려진 위협 및 알려지지 않은 위협에 대한 탐지와 실시간 모니터링 기능, 내부 위협에 대한 대응이 가능하다. 결국, SIEM의 핵심은 사용자 및 서비스 권한, 디렉토리 서비스 및 기타 시스템 구성 변경을 모니터링하고 도움을 제공하는 것이

며, 추가로 로그 감사/검토 및 사건 별 응답을 제공하여 다양한 보안 위협을 실시간으로 대응하는 것이다[7].

Ⅲ. SIEM 솔루션 소개 및 비교분석

본 절에서는 가트너 사에서 소비자들의 평가를 기반으로 2018년에 우수 제품으로 선정된 상위 제품 몇 가지를 설명한다. (그림 1)은 2018년에서 가트너 사에서 선정된 상위 SIEM 제품들이다[8]. 아래 그림의 'LEADERS' 는 사용자의 요구사항과 기능을 잘 반영하는 제품들이며 'CHALLENGERS' 는 소규모의 SIEM 고객들을 가지고 있으며, 요구사항의 일부분을 충족하는 제품을 제공하는 기업들로 구성된다. 'VISONARIES' 는 SIEM 시장의 요구사항과 강력한 기능을 제공하지만 'LEADERS' 보다는 다소 떨어지는 기능을 탑재한 업체들의 집합 군이다. 'NICHE PLAYERS' 는 특정 사용자와 요구 기능을 제공하는 기업들로 구성된다.

(그림 1) 2018 Magic Quadrant for SIEM

1) AlienVault - OSSIM

OSSIM은 AlienVault 사에서 오픈 소스로 제공되는 SIEM 제품이며 AlienVault USM 제품에서 로그 관리와 클라우드 모니터링을 제외한 기능이 제공된다[9]. 즉, OSSIM과 USM은 전혀 다른 제품이 아니라 USM의 일부분을 오픈소스로 공개한 제품이 OSSIM 솔루션이다. OSSIM이 제공하는 기능들은 Asset Discovery & Inventory, Vulnerability Assessment, Intrusion Detection, Behavioral Monitoring, SIEM Event Correlation이 있다. USM에 비해 지원하는 기능이 적지만 처음 보안을 접하는 사용자와 테스트베드 구성 및 SIEM 연구와 오픈 소스의 장점을 이용한 타 플랫폼과의 연동에는 적합하다. 또한, AlienVault 사의 제품들의 특징은 악성 호스트 관련 데이터 플랫폼인 OTX(Open Threat Exchange)를 이용하여 실시간으로 IP addresses, Domains, Hostnames, Email, URL, URI, File Hashes, CIDR Rules, FILE Paths, MUTEX name, CVE number와 같은 IOC(Indicators of Compromise)를 제공한다. OSSIM의 가장 큰 목적은 보안을 연구하는 연구원들이나, 기업 내에서 보안 시스템을 연습하려는 사람들에게 보안의 가시성과 네트워크 제어 능력을 갖출 수 있도록 도움을 주는 것이다.

2) AlienVault - Unified Security Management (USM)

USM은 AlienVault 사에서 제공하는 SIEM 솔루션으로 Windows 와 Mac OS에서 사용이 가능하고, 대상 사용자는 다양한 규모의 IT 기업이다[9]. OSSIM과는 다르게 오픈 소스로 제공하지 않고 Cloud-host에서 SaaS로 제공된다. USM은 Log Management와 Security Orchestration & Automation, Integration With Third-party Ticketing Software 등의 기능이 포함되어 있다. 클라우드, On-Premise, 하이브리드 환경의 인프라의 모니터링이 가능하며, Amazon의 AWS(Amazon Web Service)와 Microsoft Azure 클라우드 등의 모니터링을 지원한다는 특징이 있다. OSSIM과 마찬가지로 악성 호스트 관련 데이터 플랫폼인 OTX를 이용하여 실시간으로 정보를 얻을 수 있으며, 취약점 평가와 네트워크 및 호스트 기반의 침입 탐지 툴을 제공하는 다면적인 보안 솔루션이다.

3) LogRhythm - LogRhythm SIEM

LogRhythm SIEM은 LogRhythm 에서 제공하는 SIEM 으로 Windows와 Linux OS 환경에서 사용이 가능하며 인터페이스와 사용법에 대한 다양한 설명을 제공하여 전문 인력이 부족한 중소 규모의 기업에서 사용하기에 적합하다[10]. 자동화된 빠른 조치를 취하는 SmartResponse 기술을 사용하여 전체적인 분석 시간과 절차를 줄여 신속한 대처가 가능하다. LogRhythm SIEM은 보안 운영 센터의 탐지 및 대응 프레임워크인 TLM(Threat Lifecycle Management)를 통해 MTTD(Mean Time To Detect) 및 MTTR(Mean Time To Response)를 줄여 가용성을 높이며, 즉각 대응 워크 플로우인 SOAR(Security Orchestration, Automation and Response)로 사이버 위협을 탐지하고 위협 수준과 조사 및 재조정을 가속화하여 대응시간을 줄인다. 또한, 대표적인 Add-On 제품으로 Cloud AI와 네트워크 모니터링 툴인 NetMon 과 SysMon을 제공하여 모니터링 기능을 보완한다.

4) Splunk - Splunk Enterprise

Splunk Enterprise는 Splunk에서 제공하는 SIEM 솔루션으로 Windows와 Linux 환경 및 Docker 컨테이너로도 설치할 수 있다는 장점을 갖고 있다[11]. 보안, IT 운영, 비즈니스 분석 등의 솔루션을 제공하지만, 분석 후의 결과에 대한 평가를 얻기 위해서는 업체로 문의를 해야 하므로 대기업을 염두에 두고 설계된 플랫폼임을 알 수 있다. 연결된 디바이스와 사용하는 자원 등을 관리하는 Asset Investigator가 악의적인 행동을 표시해주며, 모든 원본의 로그, 머신 데이터, DevOps, IoT 및 기타 데이터를 수집할 수 있다. 빅 데이터를 처리할 수 있는 하둡(Hadoop)을 이용하면 품질의 향상도 가능하다. 또한, 자체 앱 스토어인 Splunkbase을 이용하여 모든 Splunk 제품에서 구동이 가능하고 서드 파티 통합, 애널리틱스, 자동화 기능 등을 추가 가능하다는 장점이 있다.

5) McAfee - Enterprise Security Manager (ESM)

McAfee ESM은 McAfee에서 제공하는 솔루션이며 Windows와 Mac OS에서 사용이 가능하다[12]. All-In-One 또는 개별적인 제품을 제공하여 공공시설 및 핵심 인프라, 의료, 교육 시설이 주 고객 층이다. ESM의 가장 큰 특징은 다양한 규모의 물리적, 가상 환경에서 사용할 수 있으며, 하이퍼바이저 및 클라우드 플랫폼을 지원한다. 또한, 특정 사용사례와 파트너 플랫폼에 대한 모니터링 및 경고가 가능한 콘텐츠 팩을 지원하고, 12개 이상이 서드 파티 공급업체와 통합 파트너쉽을 통해 ESM의 확장성을 극대화 시킨다. 36 곳 이상의 파트너와 수백 가지의 표준화된 데이터 소스, 업계의 위협 정보와 통합하여 확장 가능한 분산 설계가 가능하다. 이러한 이유로 McAfee ESM은 다른 시스템과 긴밀하게 통합 연계가 가능하여 단일 인터페이스를 통해 신속한 분류 및 문제 해결이 가능하다.

업체	솔루션	장점	단점
AlienVault	OSSIM	▪ USM의 일부분, 오픈 소스로 제공 ▪ OTX를 통해 실시간으로 악성 호스트 정보를 얻을 수 있음	▪ Log Management 및 클라우드 모니터링 등 기능이 부족
	USM	▪ Cisco Umbrella 및 Google G-Suite, Microsoft Office 365 등의 통합 플랫폼 포함 ▪ 설치가 간단하여 다양한 클라우드 기반 플랫폼과 연동 가능	▪ 중견 기업과 소규모 목표로 인해 엔터프라이즈 중심 기능 부족
LogRhythm	LogRhythm SIEM	▪ SCADA, IT 또는 OT(Operational Technology) 환경의 보안 이벤트 모니터링과 병합 가능 ▪ UX 및 UI 요소와 사용 용이성에 중점 ▪ 독립적인 옵션을 가진 단일 벤더 방식 제공	▪ 앱스토어 제공이 미흡하며 서드 파티에게 API의 제공이 상대적으로 부족
Splunk	Splunk Enterprise	▪ Enterprise Security 와 User Behavior Analytics 추가 가능 ▪ PII(Personally Identifiable Information) 보호 기능과 난독화, 마스킹 기능이 강력	▪ On-Premise 환경에서 적용이 부적합하며, FIM(Functional Independence Measure) 및 EDR(Endpoint Detection & Response)에 대한 미지원
McAfee	McAfee ESM	▪ Apache Kafka 및 Elasticsearch와 같은 빅 데이터 플랫폼 활용 ▪ ESM(Enterprise Security Management), MI(Management Information)를 사용하는 대기업이 사용하기에 적합	▪ 고유한 UEBA(User & Entity Behavior Analytics)가 없으며 화이트 라벨 파트너쉽을 사용
SolarWinds	LEM	▪ Windows 10을 위한 에이전트 업데이트 ▪ SSO(Single Sign-On), 관리 콘솔 업데이트 및 향상된 보안 시스템 서비스 제공	▪ 폐쇄형 시스템으로 통합이 어렵고, SNMP(Simple Network Management Protocol)와 전자메일의 일방적인 연결만 가능 ▪ 공용 클라우드 서비스인 IaaS 및 SaaS 모니터링 지원 부족

(표 1) 각 SIEM 제품들의 장/단점 비교

6) SolarWinds - Log & Event Manager (LEM)

SolarWinds Log & Event Manager는 SolarWinds에서 제공하는 SIEM으로 간단한 아키텍처와 상관관계 규칙 라이브러리를 제공하여 소기업과 중소기업이 주 고객층이다[13]. Windows 이벤트 로그를 활용하여 광범위한 로그 관리 기능과 보고기능을 가지고 있다. HIPPA, PCI DSS, SOX, ISO 등의 규정 준수를 위한 보고서 명세들을 포함하고 있고 USB 디바이스 모니터링, 파일 무결성 모니터링, 고급 탐색과 포렌식 분석이 가능하다. 또한, 데이터 쿼리를 기다리지 않고 이벤트 시간 알림 및 수정을 위한 In-Memory, 크로스 플랫폼 이벤트 처리를 지원한다. 무엇보다 LEM의 가장 큰 장점은 타 플랫폼들과 다른 대시보드의 디자인으로, 시각화 툴의 간결함은 사용자의 분석에 도움을 준다.

(표 1)은 6개의 제품에 대한 장점과 단점을 정리해 놓은 표이다. 기존 SIEM의 경우 강력하지 않은 상관 분석, 사용의 어려움, 최신의 경향과 분석이 부진함, 규칙 기반의 접근방식과 고비용의 문제를 가지고 있었다. 또한, 처음으로 보안 정책을 마련하려는 기업이나 사용자는 어떤 솔루션이 클라이언트 환경에 가장 적절한지 위의 문제들을 생각하면서 이용하기에 어려움이 있었다. 대부분의 제품들은 기존 SIEM에서 보안 시장과 사용자의 요구에 맞추어서 발전해 왔다. 이를 바탕으로 SIEM을 적용하려는 사용자나 기업은 제품마다 장점과 단점, 지원하는 기능이 다르므로 충분한 비교와 확인을 통해 시스템에 적용해야 한다.

Ⅳ. OSSIM 활용 사례 및 방안 제안

실제로 오픈 소스로 제공되는 OSSIM을 이용하여 중요한 산업 시설에서 SCADA나 다른 보안 플랫폼과 연동하는 연구가 활발히 진행되고 있다. 대표적으로 OSSIM을 이용한 새로운 보안 시스템 개발, 댐의 프로세스와 작동을 모니터링 하는 시스템, SCADA 허니팟과 OSSIM의 연동, 클라우드 컴퓨팅 모니터링 등의 사례가 있다[14, 15, 16, 17, 18, 19, 20].

위의 대표적인 사례들처럼 소스 코드나 ISO 파일을 수정하여 재설계를 한다면 작업하는 환경에 맞추어 사용이 가능하다. OSSIM을 이용하여 얻을 수 있는 이점에는 상업적인 플랫폼을 이용하는 것보다 유지, 보수, 관리비를 크게 절감할 수 있다. 또한, 상업적인 SIEM 제품 사용하기 전에 먼저 접함으로써 학습이 가능하고, 산업 보안 환경에서 시뮬레이션이 가능하다.

논문에서 제안하는 OSSIM의 활용 방안 중 하나는 기존의 SCADA 허니팟을 구성하는 시스템과 SDN(Software Defined Network)/NFV(Network Functions Virtualization) 기반의 OSSIM과 연동하여 다양한 로그 자료와 이벤트 데이터 등을 분석하는 방법을 제안한다. SDN/NFV는 네트워크 기능을 추상화하여 표준화된 컴퓨팅 노드의 소프트웨어를 통해 OpenFlow를 이용하여 네트워크의 트래픽 전달 동작을 소프트웨어 기반 컨트롤러에서 제어, 관리하는 방식이다. OpenFlow는 네트워크 스위치나 라우터의 포워딩 계층과 제어 계층 기능을 분리하여 통신하는 프로토콜이다. SDN/NFV 기술의 장점은 데이터 플레인과 컨트롤 플레인을 분리하여 네트워크의 세부 구성정보에 얽매이지 않고 요구사항에 따라 네트워크를 관리할 수 있다. 또한, 유지 보수비용을 절감할 수 있고 사용 중인 자원들을 효율적으로 관리할 수 있다[21].

SCADA 테스트베드 시스템은 통신 장애와 Spoofing, DDoS, Session Hijacking, Sniffing 같은 공격에 대한 모의가 가능하도록 설계한다. 테스트베드 시스템에 SDN/NFV 기반의 OSSIM을 이용하여 실시간으로 유입되는 트래픽 부하를 낮추고, 시스템 자원을 효율적으로 사용하여 APT나 랜섬 웨어, Zero-Day 공격 등 악성코드의 행동, 공격 동향을 분석, 탐지한다. 제안된 방법을 이용하면 향후 SCADA 같은 주요 설비에 대한 보호와 악의적인 침입이나 해킹뿐 만 아니라 IoT 등에 대한 보안성을 높이며, 최신 악성코드들에 대한 동향을 파악하고, 보안 장비 제조사들의 장비 개발과 네트워크 보안에 대한 연구에 활용할 수 있을 것으로 기대된다.

Ⅴ. 결론

IT 기술이 발전함에 따라 네트워크도 급속도로 성장하였지만, 이로 인해 보안에 대한 문제도 끊임없이 발생하고 있다. 또한, SCADA와 같은 산업 시스템이나, 기업에서도 여러 최신 보안 위협도 발생하며, 보안 사고로 인한 피해액도 증가하고 있다. 그에 반해 이런 문제들을 관리할 수 있는 보안 인력들은 턱없이 부족하고, 빠르고 더 강력하게 확산되는 보안 위협들을 사람이 모두 처리하기에는 역부족인 상황이다. 인력 부족과 진화하는 보안 위협을 생각한다면 SIEM의 도입은 보안을 위해서라면 필수적으로 행해져야 한다. SIEM 솔루션을 선택할 시 분석, 개발 및 유지 관리가 가능한 충분한 자원을 갖추지 못한 조직은 상용 기술에 비해 저렴한 오픈 소스인 OSSIM과 Elasticsearch, Logstash 및 Kibana, Apache Spot과 같은 대형 데이터 플랫폼과 통합하여 예산을 줄이고 보안 관리의 효율을 높일 수 있다. 따라서 SIEM 솔루션을 사용할 때에는 기본 기능 대비 고급 기능의 상대적인 중요성, 예산, 배포 규모, 제품의 복잡성, IT 조직의 프로젝트 배포 및 기술 지원 기능, 기존의 어플리케이션, 데이터 모니터링 및 관리 인프라와 통합 같은 조직 별 요구 사항에 따라 명확한 목표를 잡고 도입 목적과 업무 특성을 파악한 후 구축을 진행해야 한다.

ACKNOWLEDGMENT

본 연구는 한국전력공사의 2018년 착수 에너지 거점대학 클러스터 사업에 의해 지원되었음. (과제번호:R18XA05)

참 고 문 헌

[1] Saurabh Vadiya, Prashant Ambad and Santosh Bhosle, "Industry 4.0 - A Glimpse," In *proc. of 2nd International Conference on Materials Manufacturing and Design Engineering*, pp. 233-238, 2018.

[2] Yuan Gao, Xin Xie, Mithil Parekh and Edita Bajramovic, "SIEM: Policy-based Monitoring of SCADA Systems," In *proc. of Informatik 2016*, pp. 559-570, 2016.

[3] Gartner, "Reviews for Security Information and Event Management," https://www.gartner.com/, Jul. 2018, Retrieved Apr. 2019.

[4] Jong Heyon Kim, Seon Hui Im, Ik Kyun Kim, Hyun Sook Cho and Byung Kyu No, "Technical Trends of Cyber Security with Big data," *Journal of 2013 Electronics and Telecommunications Trends*, Vol. 28, No. 3, pp. 19-29, Jun. 2013.

[5] Jong-Wouk Kim, Ji Won Bang and Mi-Jung Choi, "Development Trend of SIEM for Cyber Security," *Journal of Korea Information Processing Society*, Vol. 25, No. 2, pp. 208-211, Nov. 2018.

[6] Jae-Hwa Sim, Sung-Hwan Kim and Tai-Myoung Chung, "A Survey of Solutions using Security Information Event Management," In *proc. of Korea Institute of Communication Sciences*, pp. 390-391, Jan. 2014.

[7] ByungRae Cha, MyeongSoo Choi, EunJu Kang, Sun Park and JongWon Kim, "Trends of SOC & SIEM Technology for Cybersecurity," *Journal of Smart media journal*, Vol. 6, No. 4, pp. 41-49, Dec. 2017.

[8] Kelly Kavanagh, Toby Bussa and Gorka Sadowski, "2018 Magic Quadrant for Security Information and Event Management," https://www.gartner.com/, Dec. 2018, Retrieved Apr. 2019.

[9] AT&T Cybersecurity AlienVault, https://alienvault.com/, Retrieved Apr. 2019.

[10] LogRhythm, https://logrhythm.com/, Retrieved Apr. 2019.

[11] Splunk, https://www.splunk.com/, Retrieved Apr. 2019.

[12] McAfee, https://www.mcafee.com/, Retrieved Apr. 2019.

[13] SolarWinds, https://www.solarwinds.com/, Retrieved Apr. 2019.

[14] Muhammad Masood Anwar, Muhmmad Faisal Zafar and Zafar Ahmed, "A Proposed Preventive Information Security System," In *proc. of International conference on Electrical Engineering on IEEE*, pp. 1-6, Apr. 2007

[15] Luigi Coppolino, Salvatore D' Antonio, Valerio Formicola and Luigi Romano, "Integration of a System for Critical Infrastructure Protection with the OSSIM SIEM Platform: A dam case study," In *proc. of International Conference on Computer Safety, Reliability, and Security*, pp. 199-212, Sep. 2011.

[16] Athar Mahboob and Junaid Ahmed Zubairi, "Securing SCADA Systems with Open Source Software," In *proc. of High Capacity Optical Networks and Emerging/Enabling Technologies*, pp. 193-198, Dec. 2013.

[17] Pasquale Puzio and Sergio Loureiro, "Elastic SIEM: Elastic Detector integrated with OSSIM," In *proc. of 8th International Conference on Availability, Reliability and Security*, Sep. 2013.

[18] Arthur Jicha, Mark Patton and Hsinchun Chen, "SCADA Honeypots: An In-depth Analysis of Conpot," In *proc. of 2016 IEEE Conference on Intelligence and Security Informatics*, pp. 196-198, Sep. 2016.

[19] Briffaut Jeremy, Jean-Francois Lalande and Christian Toinard, "Security and Results of a Large-Scale High-Interaction Honeypot," *Journal of computers*, Vol. 4, No. 5, pp. 395-404, May. 2009.

[20] Jong-Joo Lee, Seog-Joo Kim and Dong-Joo Kang, "A SCADA Testbed Implementation Architecture for Security assessment," *Journal of Korean Institute of Illuminating and Electrical Installation Engineers*, Vol. 24, No. 4 pp. 50-56, Apr. 2010.

[21] Yong Li and Min Chen, "Software-Defined Network Function Virtualization A Survey," *Journal of IEEE Access*, Vol. 3, pp. 2542-2553, Dec. 2015.

Manual Unpacking을 위한 Anti-Debugging 무력화에 관한 연구

김종욱, 방지원, 최미정
강원대학교
{goldbear564, jiwonbang, mjchoi}@kangwon.ac.kr

A Study on Automatic Disabling of Anti-Debugging in Manual Unpacking

Kim Jong Wouk, Ban Ji Won, Choi Mi Jung
Kangwon National Univ.

요 약

최근 네트워크 확장이 급속도로 진행됨에 따라 피싱, 봇넷, DDoS 등의 보안에 대한 악성코드 기반 공격도 비례하여 동일하게 증가하고 있다. 이러한 위협을 예방하기 위해 백신 프로그램 개발 및 업데이트가 지속적으로 이루어지고 있으나, 악성코드 또한 날이 갈수록 발전되어 분석가들의 분석 행위를 방해하고, 악성코드의 수명을 늘려 더욱 큰 피해를 불러일으킨다. 대표적으로 악성코드 분석 및 탐지를 방해하는 방법 중 하나인 Anti-Debugging 기술은 소프트웨어를 보호하는 기술 중 하나이지만, 악성코드 제작자들은 이를 악용하여 자신들의 악성코드를 보호하는 목적으로 사용하고 있다. 본 논문에서는 위와 같이 악성코드를 보호하고 은닉하는 기술인 Anti-Debugging 기술들에 대해 소개하고, Anti-Debugging 우회하는 방안을 제시한다.

Ⅰ. 서 론

최근까지 바이러스, 키로거와 같은 악성코드 기반 소프트웨어로 인해 다양한 방법으로 시스템에 접근하여 사용자의 행위, 개인정보 및 금융 정보뿐만 아니라 기업, 공공기관의 중요 데이터 탈취, 손상과 같은 치명적인 피해가 지속적으로 발생하고 있다. 악성코드로 인한 피해는 사용자의 금융 정보와 같은 중요한 데이터의 손실에서 심각한 문제에 이르기까지 다양한 형태가 될 수 있다. Symantec 사의 ‘Internet Security Threat Report 2019’ 보고서에 따르면 특정 악성 소프트웨어는 2018년도에 많은 악성코드를 확산시켜 금융 트로이 목마를 16%나 차지했다고 한다[1]. 또한, Trend Micro 사의 ‘Unseen Threats, Imminent Losses’ 보고서에는 산업시스템을 구성하는 요소 중 하나인 Supervisory Control And Data Acquisition(SCADA) 시스템에 대한 취약점이 발견되었으며, 악성코드를 사용하여 특정 운영에 대한 진단데이터를 수집 및 SCADA 시스템을 제어할 수 있다고 보고하였다[2]. 위와 같은 보고서를 통해서 악성코드로 인한 피해는 꾸준히 증가하고 있다는 것을 알 수 있다.

보안 전문가들은 악성코드로 인한 피해를 최소화하기 위해 백신 프로그램과 보안 장비의 배치와 같은 다양한 방법을 통해 보안을 강화하고, 분석가들은 디버거라는 소프트웨어를 사용하여 웜과 같은 바이러스를 신속하게 분석하고 퍼지지 않도록 처리하는 등의 노력을 기울이고 있다[3]. 하지만, 악성코드 배포 및 제작하는 해커들도 분석 행위를 방해하기 위해 Anti-Debugging, Packing과 같은 Anti-Analysis 기술 등을 활용하고 있어 악성코드 분석에 어려움을 겪고 있다. 분석 대상인 디버기, 즉 악성코드에 Anti-Analysis 기술들이 적용되어 있다면, 디버거 프로그램으로 탐지하지 못할 가능성이 존재하고, 이를 해결하기 위한 분석가의 노력과 많은 시간이 필요하다.

본 논문에서는 디버거의 취약점을 공격하여 출력 및 입력장치의 신호를 무시하거나 디버거를 종료시키는 등의 Anti-Analysis 기술 중 대표적으로 사용하는 Anti-Debugging 기술들을 소개하고, Anti-Debugging 기술이 어떻게 디버거가 디버기로부터 숨겨지는지에 대해 설명하며, Anti-Debugging 기술을 무력화하는 방법을 제시한다.

Ⅱ. Background and Related Work

1) Anti-Debugging

Anti-Debugging 기술은 컴파일된 코드의 지적 재산권을 보호하기 위해 개발되었지만, 악성코드 제작자들은 악성코드의 수명을 늘리기 위해 사용한다. 디버깅(Debugging)이란 프로세스가 예기치 않게 동작하거나 충돌하여 발생하는 에러, 버그 및 오류를 감지하여 제거하는 과정을 의미하며 또한, 디버깅은 악성코드를 동적으로 분석하는 과정에서도 사용이 된다[4]. 분석가들은 디버깅을 수행할 때 Ollydbg, IDA Pro와 같은 디버거를 사용하여 분석하지만, 디버거는 Self-Modifying, 키보드, 마우스와 같은 입력 장치의 이벤트를 차단하는 기술에 매우 취약하여 악성코드는 디버깅 여부를 판단하여 분석을 지연시킨다[5, 6].

Anti-Debugging에 관한 연구는 이전부터 지속적으로 이루어졌다. Tyler는 Anti-Debugging 기술을 사용하여 프로그램을 개발하는 방법에 관한 연구를 소개하였으며, Branco는 Anti-Debugging, Anti-Virtual Machine, 난독화 기술 등 악성코드가 사용하는 분석 방해 기술에 대한 개요를 제시하였다[7, 8]. Shang은 Windows의 디버깅 매커니즘에 대한 개요와 디버거가 사용하는 디버깅 방법에 따른 분류와 디버깅 함수 및 예외를 사용하는 Anti-Debugging에 관하여 설명한다[9]. 이러한 Anti-Debugging 기술들을 무력화하기 위해 Lee는 Anti-Debugging Rule-Set을 정하여 분석 대상에서 부합되는 부분을 다른 명령어로 치환하는 방법을 제안하였으며, Hao는 Apate라는 프레임워크를 제안하여 Anti-Debugging 기술을 17개의 카테고리로 나누어 무력화하는 연구를 진행하였다[10, 11, 12].

2) 패킹(Packing)

패킹이란 ‘Executable Compression’라고도 불리며, 실행 파일의 크기를 줄여주는 압축 기술 중 하나로, 실행 파일의 형태를 유지하고 압축을 통해 파일의 크기를 줄여 저장 공간을 확보하기 위해 개발된 기술이었으나, 악성코드 제작자들은 악성코드의 은닉을 위해 사용하였다[13]. WildList 사의 2006년 보고서에 따르면 92% 이상의 악성코드에 패킹 기술이 적용되어 있다는 것을 알 수 있다[14]. 패킹된 파일은 기존의 코드를

변형시키기 때문에, 패킹된 악성코드를 분석하기 위해서는 반드시 패킹을 해제하는 언패킹 작업을 수행해야 한다. 패킹 여부를 판단하기 위해서는 Choi는 PE 파일의 헤더를 분석하여 패킹 여부를 판단하는 PHAD를 제안하였다. PHAD는 파일의 PE 헤더의 특징을 선정할 때 일반 파일과 패킹된 파일의 변수를 heuristic analysis를 통해 8개를 선정하고, 이를 기반으로 패킹 여부를 탐지한다[15]. Jenong은 패킹된 악성코드를 언패킹 하는 것은 압축 알고리즘을 모르더라도 동적으로 가능하며, 동적으로 언패킹 하는 과정에서 엔트로피 값의 변화량을 측정하여 압축 알고리즘을 다수의 클러스터로 분류하는 방법을 제안하였다[16].

Ⅲ. Anti-Debugging 기술 소개 및 우회 방법

디버거의 분석이 시작되면 디버기와 상호작용할 수 있도록 운영체제에 의해 디버기의 환경이 변경된다. Anti-Debugging 기술은 변경된 환경 데이터를 참조하여 디버깅 여부를 판단할 수 있고, 디버깅 중이라면 분석을 종료시킬 수 있다. 이와 같은 공격을 해결하기 위해 본 절에서는 Windows에서 제공하는 함수를 사용하여 디버깅 여부를 판단하는 방법, 수정된 데이터를 탐지하여 디버깅 여부를 판단하는 방법 등의 Anti-Debugging 기술에 대해 소개하고, Anti-Debugging 기술을 무력화하는 방법을 설명한다.

1) IsDebuggerPresent

IsDebuggerPresent 함수는 소프트웨어가 가장 쉽게 디버거의 존재를 탐지할 수 있는 함수로써, 대부분이 악성코드에 사용되며, 이 함수는 UPX, PECompact와 같은 대부분의 패커에 포함된 Anti-Debugging 함수이다. 또한, Process Environment Block(PEB) 구조체의 BeingDebugged 필드를 참조하여 디버거가 탐지될 경우 TRUE를 반환한다[17, 18, 19].

(그림 1)의 좌측 코드는 IsDebuggerPresent의 내부 함수의 원형으로, PEB 구조체의 BeingDebugged 멤버를 확인하여 반환한다. 이 함수를 우회하기 위해서는 항상 0x0으로 반환하도록 코드를 수정이 필요하다. 본 논문에서는 SUB 명령어를 이용하여 EAX 레지스터에 0x0이 저장되게 만들어 우회하였다. 이때 EAX 레지스터에 0x0을 저장하여 반환하도록 유도하는 것이 가장 중요하다.

```
MOV EAX,DWORD PTR FS:[18]      MOV EAX,DWORD PTR FS:[18]
MOV EAX,DWORD PTR DS:[EAX+30]  SUB EAX,EAX
MOVZX EAX,BYTE PTR DS:[EAX+2]  RETN
RETN
```

(그림 1) IsDebuggerPresent 원본 코드(좌)와 수정 코드(우)

2) CheckRemoteDebuggerPresent

CheckRemoteDebuggerPresent 함수는 Windows NT 버전 이상에서 사용이 가능한 함수이며, 특정 프로세스의 Process Identifier(PID)를 전달하면 그 프로세스가 디버깅 여부를 판단한다. 디버거로 내부 함수를 실행하면 NtQueryInformationProcess함수를 호출하기 때문에, NtQueryInformationProcess 함수를 우회할 경우, CheckRemoteDebuggerPresent 함수도 우회가 가능하다. 또한, 몇몇 패커들은 CheckRemoteDebuggerPresnet 함수를 사용하여 디버깅을 방지하거나 직접적으로 NtQueryInformationProcess함수를 호출하여 디버깅을 방지한다[20].

(그림 2)의 좌측 코드는 CheckRemoteDebuggerPrsent 함수의 일부 생략된 내부 코드이다. CheckRemoteDebuggerPrsent 함수를 우회하기 위해서는 NtQueryInformationProcess 함수를 우회하거나, 그 전에 EAX 레지스터에 0x0을 저장하여 함수를 반환하여 종료하면 된다. 본 논문에서는 (그림 2)의 우측 코드와 같이 NtQueryInformationProcess함수를 호출하기 전에, PUSH 명령어로 0x0을 스택에 저장하여 POP 명령어로 EAX 레지스터에 0x0을 저장한 후 RETN 연산으로 종료를 함수를 종료한다. 이때 중요한 것은 EAX 레지스터를 0x0으로 만든 후, 스택에 저장한 EBP 레지스터의 값을 POP 명령어로 꺼낸 다음 함수를 끝내는 것이 중요하다.

```
MOV EDI,EDI                          MOV EDI,EDI
PUSH EBP                             PUSH EBP
MOV EBP,ESP                          MOV EBP,ESP
CMP DWORD PTR SS:[EBP+8],0           MOV EAX,DWORD PTR SS:[EBP+C]
PUSH ESI                             PUSH 0
JE SHORT kernel32.76C13FC1           POP DWORD PTR DS:[EAX]
MOV ESI,DWORD PTR SS:[EBP+C]         XOR EAX,EAX
TEST ESI,ESI                         POP EBP
JE SHORT kernel32.76C13FC1           RETN 8
...                                  ...
POP ESI                              POP ESI
POP EBP                              POP EBP
RETN 8                               RETN 8
```

(그림 2) CheckRemoteDebuggerPrsent 코드(좌)와 수정 코드(우)

3) NtQueryInformationProcess(ZwQueryInformationProcess)

ZwQueryInformationProcess 함수는 시스템 호출의 Wrapper 함수이다. (그림 3)은 ZwQueryInformationProcess 함수에 대한 파라미터 값을 나타낸 것이다. 파라미터 값 중 ProcessInformationClass 파라미터는 검색할 프로세스의 정보 중 어떤 정보을 수집할지에 대한 정보를 나타내는 값으로, 열거형 데이터인 ProcessInfomationClass의 ProcessDebugPort를 의미하는 0x7 값을 설정하여 이 함수를 호출하여 디버깅 여부를 알 수 있다[21]. 만약, 함수 호출 후 결과값이 0xFFFFFFFF(-1)라면 디버거가 탐지된 경우이며, 0x0이 반환된 경우는 디버거가 탐지되지 않은 경우를 뜻한다.

(그림 4)의 좌측 코드는 ZwQueryInformationProcess의 원본 코드이며, 이 함수를 우회하기 위해서 수정한 코드는 (그림 4)의 우측 코드이다. 이 함수의 수정을 할 때, 원본 코드의 2번째 코드까지는 반드시 수행되어야 하고 0x0 값을 반환해야 한다. 만약 ProcessInformationClass가 ProcessDebugPort(0x07)와 같다면 0x0 값을 PUSH 명령어로 스택에 저장하고, POP 명령어로 스택에 저장한 데이터를 EAX에 가져와 0x0을 반환하여 우회를 할 수 있다.

```
NTSTATUS WINAPI ZwQueryInformationProcess(
  _In_      HANDLE           ProcessHandle,
  _In_      PROCESSINFOCLASS ProcessInformationClass,
  _Out_     PVOID            ProcessInformation,
  _In_      ULONG            ProcessInformationLength,
  _Out_opt_ PULONG           ReturnLength
);
```

(그림 3) ZwQueryInformationProcess의 파라미터

```
MOV EAX,0EA                          MOV EAX,0EA
MOV EDX,7FFE0300                     MOV EDX,7FFE0300
CALL DWORD PTR DS:[EDX]              CMP DWORD PTR SS:[ESP+8],7
RETN 14                              JE SHORT 003F001D
                                     PUSH 775E6052
                                     RETN
                                     MOV EAX,DWORD PTR SS:[ESP+C]
                                     PUSH 0
                                     POP DWORD PTR DS:[EAX]
                                     XOR EAX,EAX
                                     RETN 14
```

(그림 4) ZwQueryInformationPrcoess 원본 코드(좌)와 수정 코드(우)

4) Process Environment Block(PEB)

Process Environment Block은 프로세스마다 할당되어 있으며 프로세스의 정보를 담고 있는 구조체이다. 악성코드는 함수를 사용하지 않고, 디버깅 여부를 판단할 수 있는데, 이때 사용되는 것이 PEB 구조체이며, 뿐만 아니라 PECompact, ASPack, ASProtect와 같은 많은 패커에서도 사용이

된다[22, 23, 24].

PEB구조체의 BeingDebugged, Process Heap,NtGlobalFlag 멤버를 확인하여 Anti-Debugging 기능을 수행할 수 있다. BeingDebugged 멤버는 디버깅 중에 TRUE(non-zero) 값으로 설정이 되며 디버깅 중이 아니라면 FALSE(zero) 값으로 설정된다. NtGlobalFlag 멤버는 디버깅 중이 아닐 때, 0x0 값으로 설정되지만, 디버깅 중일 때에는 0x70 값으로 설정된다. (표 1)은 NtGlobalFlag의 Flag 구성을 나타낸 것이다[25].

Flag	Value
FLG_HEAPENABLE_TAIL_CHECK	0x10
FLG_HEAP_ENABLE_FREE_CHECK	0x20
FLG_HEAP_VALIDATE_PARAMETER	0x40

(표 1) NtGlobalFlag 멤버의 Flag 구성

PEB 구조체 중 HEAP 구조체를 사용하여 디버깅 여부를 판단할 수 있다. Anti-Debugging에 사용되는 HEAP 구조체의 멤버는 Flags, ForceFlags 멤버가 사용된다. Flags 멤버는 디버깅 중이 아니라면 0x2 값으로 설정되며, 디버깅 중이라면 0x50000062 값으로 설정된다. ForceFlags 멤버는 디버깅 중이 아니라면 0x0 값으로, 디버깅 중이라면 0x40000060 값으로 설정된다. (표 2)와 (표 3)은 Flags와 Force Flags의 Flags 구성을 각각 나타낸 것이다.

Flag	Value
HEAP_GROWABLE	0x2
HEAP_TAIL_CHECKING_ENABLED	0x20
HEAP_FREE_CHECKING_ENABLED	0x40
HEAP_SKIP_VALIDATION_CHECKS	0x10000000
HEAP_VALIDATE_PARAMETERS_ENABLED	0x40000000

(표 2) Flags 멤버의 Flag 구성

Flag	Value
HEAP_TAIL_CHECKING_ENABLED	0x20
HEAP_FREE_CHECKING_ENABLED	0x40
HEAP_VALIDATE_PARAMETERS_ENABLED	0x40000000

(표 3) ForceFlags 멤버의 Flag 구성

PEB 구조체는 프로세스가 실행되는 동안 FS 레지스터 Thread Environment Block(TEB)의 시작 주소를 가지고 있으며, TEB의 0x30 오프셋 위치에 PEB 구조체를 가리키는 포인터가 있다. 이 주소를 사용하여 PEB 구조체에 접근할 수 있다.

이러한 Anti-Debugging 기술들을 우회하기 위해서는 악성코드가 PEB 구조체를 확인하여 디버깅 여부를 판단하기 이전의 PEB 구조체를 미리 수정하는 방법으로 Anti-Debugging을 무력화할 수 있다. 하지만 악성코드가 어느 시점에서 PEB 구조체를 확인하는지 알 수 없으므로 분석을 시작할 때, PEB 구조체를 OS 상에서 실행할 때와 같은 데이터를 가지고 있도록 데이터 값을 수정해서 Anti-Debugging을 무력화시킬 수 있다.

5) GetCurrentProcessId & BlockInput

이 두 함수는 디버깅 여부를 탐지하는데 사용되는 함수는 아니지만, 악성코드 또는 Yoda' Protector와 같은 상용 패커가 디버거를 공격할 때 자주 사용하는 함수들이다[26]. GetCurrentProcessId 함수는 호출 프로세스의 PID를 반환하는 함수이며, 자신을 시작한 프로세스가 자기의 PID와 동일한 PID를 가졌는지 확인한다. 즉, 디버거를 통해 시작되었는지 판단하여 만약 PID가 다르다면 디버거를 종료시키기 때문에, 이 함수를 우회하기 위해서는 디버거의 PID를 반환한다면, 우회가 가능하다. (그림 5)의 좌측 코드는 GetCurrentProcessId 함수의 원본 코드이며, 우측 코드는 우회하기 위한 수정된 코드이다. GetCurrentProcessId 함수의 코드를 수정하기 위해서는 EAX 레지스터에 디버거의 PID를 저장하여 반환하도록 유도하는 방법으로 우회할 수 있다.

BlockInput 함수는 키보드와 마우스 입력 이벤트를 차단하는 함수이다. 만약 이 함수가 실행된다면, 분석가의 키보드와 마우스가 작동되지 않아 컴퓨터를 재시작하여야 한다. (그림 6)의 좌측 코드는 BlockInput 함수의 원본 코드이며, 우회하는 방법으로는 (그림 6)의 우측 코드와 같이 아무 동작도 수행하지 않는 코드인 NOP 명령어들로 수정하여 BlockInput 함수를 무력화하는 방법으로 우회가 가능하다.

```
MOV EAX,DWORD PTR FS:[18]
MOV EAX,DWORD PTR DS:[EAX+20]
RETN
```

```
MOV EAX,918
NOP
NOP
NOP
RETN
```

(그림 5) GetCurrentProcessId 코드(좌)와 수정 코드(우)

```
MOV EAX,1141
MOV EDX,7FFE0300
CALL DWORD PTR DS:[EDX]
RETN 4
```

```
NOP
NOP
NOP
...
NOP
NOP
NOP
RETN 4
```

(그림 6) BlockInput 코드(좌)와 수정 코드(우)

Ⅳ. 결론 및 향후 연구

악성코드 및 대부분의 패커들은 다양한 Anti-Debugging 기술을 이용하여 프로그램을 보호한다. 이러한 Anti-Debugging 기술에 잘 적응할 수 있는 진화된 디버거가 분석가들을 위해 필요하다. 그러므로, 본 논문에서는 디버거 프로그램에서 Anti-Debugging을 무력화시키는 방법을 제안하였다. 대부분의 Anti-Debugging 기술은 특정 함수의 코드를 수정하여 여러 번 호출되더라도, 한 번의 수정으로 무력화가 가능하고, 디버거 프로그램 환경을 운영체제에서 실행시킨 것과 같은 환경으로 조성하고 실험을 통해 Anti-Debugging 기술을 무력화할 수 있다는 것을 확인하였다. 그러나, 악성코드 및 패커들은 본 논문에서 언급한 Anti-Debugging 기술 이외의 다른 기술들 또한 존재하기 때문에 분석가들은 이러한 기술을 우회할 수 있는 전문 지식과 많은 시간이 필요하다. 이를 위해 향후 연구로는 Anti-Analysis 기술이 적용된 악성코드를 분석하는 비용과 시간을 단축하고, 유연하게 대처가 가능한 플러그인을 개발할 예정이다.

ACKNOWLEDGMENT

이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2017-0-00158, 국가 차원의 침해사고 대응을 위한 사이버 위협 인텔리전스 분석(CTI) 및 정보 공유 기술 개발)

참 고 문 헌

[1] Symantec, "Internet Security Threat Report 2019", from https://www.symantec.com/, Feb, 2019, Retrieved Mar, 3, 2019.

[2] Trend Micro, "Unseen Threats, Imminent Losses," from

https://www.trendmicro.com, Retrieved Mar, 3, 2019.

[3] Michael N. Gagnon, Stephen Taylor, and Anup K. Ghosh, "Software protection through anti-debugging," IEEE Security & Privacy, pp.82-84, May, 2007.

[4] Jong Wouk Kim, Ji Won Bang, and Mi Jung Choi, "A Study on analysis method of malicious code with analysis avoidance technology," In *Proc. of* KNOM 2018, pp. 21-24, May, 2018.

[5] OllyDbg, from www.ollydbg.de/, Retrieved Jan, 03, 2019.

[6] IDAPro, from https://www.hex-rays.com/products/ida/, Retrieved Jan, 12, 2019.

[7] Shields, Tyler, "Anti-debugging-a developers view," Veracode Inc., USA, 2010.

[8] Branco, Rodrigo Rubira, Gabriel Negreira Barbosa, and Pedro Drimel Neto, "Scientific but Not Academical Overview of Malware Anti-Debugging, Anti-Disassembly and Anti-VM Technologies," In *Proc. of* Black Hat Conference, Jul, 2012.

[9] Gao, Shang, and Qian Lin., "Debugging classification and anti-debugging strategies," In *Proc. of* Fourth International Conference on Machine Vision (ICMV 2011), Jan, 2011.

[10] JaeKeun Lee, BooJoong Kang, and EulGyu Im, "Rule-based Anti-anti-debugging System," In *Proc. of* the 2013 Research in Adaptive and Convergent Systems, pp. 353-354, Oct, 2013.

[11] JaeKeun Lee, BooJoong Kang, and EulGyu Im, "Evading anti-debugging techniques with binary substitution," *Journal of International Journal of Security & its Applications*, pp. 183-192.

[12] Shi Hao, and Jelena Mirkovic. "Hiding debuggers from malware with apate," In *Proc. of* the Symposium on Applied Computing, pp. 1703-1710, Apr, 2017.

[13] Silvio Cesare, Yang Xiang, and Wanlei Zhou, "Malwise—an Effective and Efficient Classification System for Packed and Polymorphic Malware," IEEE Trans. on Computers, Vol. 62, No. 6, pp. 1193-1206, Jun, 2013.

[14] Wei Yan, Zheng Zhang, and Ansari Nirwan, " Revealing Packed Malware," IEEE Security and Privacy, Vol. 6, No. 5, pp. 65-69, Oct. 2008.

[15] Yang Seo Choi, Ik Kyun Kim , Jin Tae Oh, Jae Cheol Ryou, "PE File Header analysis-based packed PE file detection technique (PHAD)," In *Proc. of* International Symposium on Computer Science and its Applications. IEEE, pp. 28-31. 2008.

[16] Guhyeon Jeong, Euijin Choo, Joosuk Lee, Munkhbayar Bat-Erdene, Heejo Lee, "Generic Unpacking Using Entropy Analysis," *Journal of Advanced Information Technology and Convergence*, Vol. 7, No. 1, pp.232-238, Feb, 2009.

[17] UPX, from https://upx.github.io/, Retrieved Feb, 21, 2019.

[18] PECompact, from https://bitsum.com/portfolio/pecompact/, Retrieved Feb, 21, 2019.

[19] Fanglu Guo, Peter Ferrie, and Tzi-Cker Chiueh., "A study of the packer problem and its solutions," In *Proc. of* International Workshop on Recent Advances in Intrusion Detection in RAID, pp. 98-115. Sept. 2008.

[20] Ferrie, Peter., "Anti-unpacker tricks-part one," Virus Bulletin 4. 2008.

[21] Microsoft, Microsoft Developer Network, from https://docs.microsoft.com/en-us/windows/desktop/procthread/zwquer yinformationprocess, Retrieved Feb, 18, 2019.

[22]Amir Afianian, Salman Niksefat, Babak Sadeghiyan, and David Baptiste, "Malware Dynamic Analysis Evasion Techniques: A Survey," arXiv preprint arXiv:1811.01190 (2018).

[22] Liţă, Cătălin Valeriu, Doina Cosovan, and Dragoş Gavriluţ. "Anti-emulation trends in modern packers: a survey on the evolution of anti-emulation techniques in UPA packers," Journal of Computer Virology and Hacking Techniques, pp. 107-126, 2018.

[23] ASPack, from http://www.aspack.com/, Retrieved Feb, 25, 2019.

[24] ASProtect, from http://www.aspack.com/, Retrieved Feb, 25, 2019.

[25] Ferrie, Peter., "The ultimate anti-debugging reference," 2011.

[26] Yoda's Protector, from https://sourceforge.net/projects/yodap/, Retrieved Feb, 22, 2019.

유해 네트워크 트래픽 탐지를 위한 컨볼루션 신경망기반 트래픽 분류 기법 연구

염성웅, 뉘엔 지앙 쓰엉, 뉘엔 반 퀴엣, 김경백
전남대학교 전자컴퓨터공학부
yeom4032yeom4032@gmail.com, truongnguyengiang.bk@gmail.com,
quyetict@gmail.com, kyungbaekkim@jnu.ac.kr

A Study on Convolutional Neural Network based Traffic Classification Methods for Detecting Malicious Network Traffic

Sungwoong Yeom, Giang-Truong Nguyen, Van-Quyet Nguyen, Kyungbaek Kim
Dept. Electronics and Computer Engineering, Chonnam National University

요 약

최근 유해 네트워크 트래픽을 탐지하기 위해 머신러닝 기법을 활용하는 방법론이 주목을 받고 있다. 이 논문에서는 딥러닝 기법중 하나인 컨볼루션 신경망 (Convolutioanl Nural Network)을 기반으로 유해 네트워크 트래픽을 분류하는 기법을 소개한다. 우선 이미지 처리에 강한 컨볼루션 신경망을 활용하기 위해, 네트워크 트래픽의 주요 정보를 규격화된 이미지로 변환하는 방법을 제안한다. 이후 네트워크 트래픽 정보를 변환한 이미지를 입력으로 컨볼루션 신경망을 학습을 시켜 제공되는 네트워크 트래픽의 분류를 수행하도록 한다. KDD 1999 데이터셋을 활용하여 이미지 변환 및 컨볼루션 신경망 기반 네트워크 트래픽 분류 기법의 성능을 검증하였다. 특히, 이미지 변환에 이용되는 트래픽 정보의 변동에 대해서 컨볼루션 신경망 기반 네트워크 트래픽 분류기법이 안정적으로 동작하는 것을 확인하였다.

Ⅰ. 서 론

네트워크 기술의 발달과 IoT 기기의 활성화에 따른 네트워크 상의 트래픽의 복잡도가 점점 높아지면서, 네트워크에 유해 트래픽을 유발시켜서 네트워크 서비스의 질을 저하시키거나 특정 서버 및 호스트의 동작에 피해를 입히는 네트워크 공격에 대한 탐지 및 방어가 더욱 중요해지고 있다. 최근, 머신러닝 기반의 네트워크 공격 트래픽 분류기법에 대한 연구가 주목을 받고 있다.[1][2] 이 연구들에서는 주로 종단간 연결정보, 도메인정보, 데이터전송정보와 같은 여러 네트워크 트래픽 정보를 특징벡터로 이용하는 다양한 분류기 (SVM, KNN, Naive Bayes)를 활용하여 네트워크 공격 트래픽을 분류하는 방법을 제안하였다.

본 논문에서는 딥러닝 기법중 하나인 컨볼루션 신경망 (Convolutional Neural Network)를 이용하여 네트워크 공격 트래픽을 분류하는 기법을 소개한다. 제안하는 기법은 KDD 1999 데이터셋에서 제공되는 네트워크 트래픽 정보를 규격화된 이미지로 변환하고, 변환된 이미지들을 이용해 컨볼루션 신경망 모델을 학습시켜 향후 네트워크 트래픽 분류를 위한 모델을 도출한다. 학습된 컨볼루션 신경말 모델을 이용해 입력되는 트래픽 정보가 일반적인 네트워크 트래픽인지 네트워크 공격에 사용되는 트래픽인지를 구분한다. KDD 1999 데이터 셋을 활용한 검증을 통해 제안되는 기법이 기존의 머신러닝 기반의 방법보다 성능이 우수함을 확인하였고, 또한 컨볼루션 신경망에 활용되는 이미지 변환 시, 특징벡터의 순서 및 그룹화가 성능에 미치는 영향이 미미함을 확인하였다.

2장에서는 관련 머신러닝 기법 및 컨볼루션 신경망에 대해 소개하고, 3장에서는 제안하는 컨볼루션 신경망 기반 유해 네트워크 트래픽 탐지 기법을 소개한다. 4장에서는 KDD 1999 데이터 셋에 기반한 제안 기법의 성능 검증 결과를 기술하고, 5장에서 본 논문의 결론 및 향후 연구 내용에 대해 기술한다.

Ⅱ. 관련연구

1. SVM (Support Vector Machine)

SVM은 주어진 특징 벡터들 간의 마진을 최대로 하는 방법을 이용해 서로 다른 특징을 가지는 데이터 집합을 분류하는 기법으로, 다수의 특징 벡터가 주어지더라도 집합간의 Support Vector를 구함으로써 분류기법을 안정적으로 운용할 수 있는 장점을 가진다.

2. KNN (K-Nearest Neighbors)

KNN은 임의의 데이터를 입력으로 이용하였을 때, 해당 데이터의 특징 벡터와 다른 데이터들의 특징 벡터와 유사도를 계산하여, 가장 유사도가 높은 K개의 데이터를 이웃으로 선택하는 기법이다. 만약 K = 1이고, KNN을 분류 기법으로 이용한다면, 입력된 데이터는 가장 유사도가 높은 하나의 그룹으로 분류된다.

3. Naive Bayes

Naive Bayes 분류 기법은 Bayes 이론을 적용하는 확률적 분류기법으로, 특징 벡터들 간의 독립성이 강할수록 그 성능이 좋아진다.

4. 컨볼루션 신경망 (Convolution Neural Network)

컨볼루션 신경망 (CNN : Convolution Neural Network) 는 영상 이미지 분류를 위한 최신의 분류 모델로, 다수의 필터를 영상 이미지의 픽셀 데이터에 적용하여 고차원 특징을 추출하여 분류기를 학습하는 모델이다. 이때, 추출되는 고차원 특징들은 convolution layer, pooling layers, fully connected layer로 구성되는 hidden layer 내부에 존재하게 되어 각 특징들에 대한 자세한 정보를 확인하기 어렵고, 특별한 의미를 부여하기 힘들다. Convolution layer는 영상 이미지의 여러 sub-region에 다양한 convolution filter를 적용하여 여러 입력을 하나의 출력으로 계산하는 비선형적 수학적 계산 모델을 적용한 계층이다. Pooling layer는

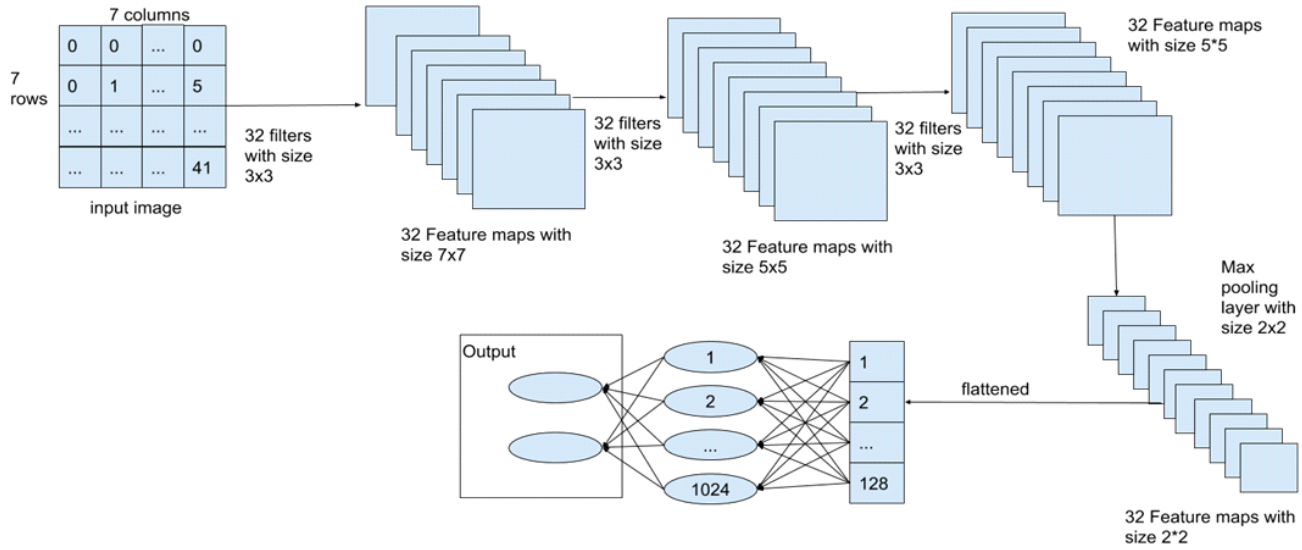


그림 2. 제안된 컨볼루션 신경망 기반 네트워크 트래픽 분류 모델 구조

convolution layer에서 계산된 데이터 계층의 크기를 줄이는 계층이고, fully connected layer는 입력되는 데이터 계층의 노드들의 값을 입력으로 사용하고 추가적인 hidden 노드들을 활용하여 출력 값을 계산하는 신경망 모델로, Supervised 또는 Unsupervised learning을 통해 해당 신경망을 학습시킨다.

Ⅲ. 컨볼루션 신경망 기반 유해 네트워크 트래픽 탐지 기법

본 논문에서는 네트워크 트래픽 정보를 이미지로 표현하고, 이를 컨볼루션 신경망을 통해 학습시켜 네트워크 공격 트래픽을 분류하는 모델을 제공하는 기법을 제안한다.

1. 네트워크 트래픽 정보의 이미지 변환

제안하는 기법을 위해 우선적으로 네트워크 트래픽 정보를 이미지로 변환하는 것이 필요하다. 우리는 KDD 1999 데이터 셋을 기준으로 네트워크 트래픽 정보를 표현하는 특징 벡터를 추출하였다. 사용되는 특징벡터는 총 41가지로 TCP connection 특징 9가지, Domain knowledge 관련 connection 특징 13가지, 2초간의 connection traffic 특징 9가지, 그리고 2초 이상에 해당하는 공격 특징 10가지를 포함한다.[3] 네트워크 트래픽 특징벡터를 이미지로 변환하기 위해, 모든 특징 백터 값을 0부터 255사이의 값으로 Normalize하고, 가로 7 픽셀, 세로 7픽셀을 가지는 정사각형 이미지로 변환하였다.

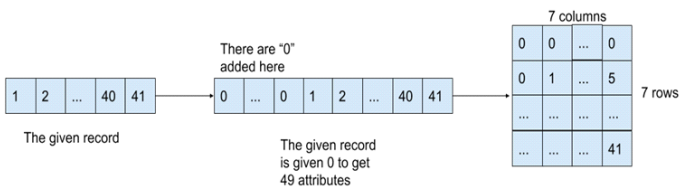


그림 1. 네트워크 트래픽 특징 벡터의 이미지 변환

특징벡터를 7X7 이미지로 변환할 때, 어떤 벡터를 이미지의 몇 번째 셀에 맵핑할지는 두가지 방법론을 사용하여 수행하였다. 첫 번째는, KDD 데이터 셋에서 제공되는 데이터 순서를 지키고 데이터 앞쪽에 0 값을 더하여 변환을 수행하는 방법이다. 두 번째 방법은 앞서 말했던 특징 그룹별로 데이터 셋을 재 정렬한 후 데이터 중간 중간에 0을 삽입하여 이미지로 변환하는 방법이다. 이러한 두 경우에 대해서 실험을 수행한 결과 공격 트래픽 탐지 결과가 유사하게 나오는 것을 확인하였다.

2. 네트워크 트래픽 분류를 위한 컨볼루션 신경망 구조

제안하는 방법에서 사용한 컨볼루션 신경망 구조는 그림 2와 같다. 7 X 7 크기의 이미지를 입력으로 사용하고, 총 3개의 convolution layer를 사용하고, 1번의 pooling을 수행 후, fully connected layer를 학습한다. Convolution layer의 경우, 3 X 3 크기의 32개 필터를 적용하여 layer를 생성하는데, 두 번째 레이어를 생성할 때는 padding을 하지 않고 convolution layer의 크기를 5 X 5로 줄인다. Pooling layer에서는 총 2x2x32 = 128개의 값을 노드로 가지게 되고, fully connected layer에서 1024개의 hidden node를 학습시킨다. 본 논문에서는 네트워크 트래픽을 일반 트래픽과 공격 트래픽 (DoS)을 구분하는 분별기를 학습시켜서 출력이 2개로 나오도록 구조를 설계하였다.

Ⅳ. 검증

제안된 기법의 성능을 검증하기 위해, KDD 1999 데이터 셋을 기반으로 Cross Validation을 수행하여, 각 분류 기법별 Accuracy와 False positive를 측정하였다. 검증을 위해 사용된 데이터 셋은 KDD 1999데이터 중 일반 네트워크 트래픽과 DoS공격 네트워크 트래픽에 대한 데이터를 선별하여 준비하였다. 전체 데이터의 90%는 분류기 모델 학습에 사용하였고, 10%는 모델 성능 테스트로 사용하였고, 10-fold Cross Validation을 수행하였다.

비교하는 분류모델로는 SVM, KNN, Naive Bayes 그리고 제안하는 CNN기반 분류모델이 있다. SVM, KNN, Naive Bayes 모델은 Weka 3.8.1을 활용하였다. [4] 제안하는 CNN기반 분류모델은 TensorFlow를 활용해 구현하였다.

그림 3에서는 네트워크 트래픽 분류기법 별 Accuracy (true detection / true attack traffic)를 나타낸다. 그림 4에서는 네트워크 트래픽 분류 기법별 False Positive (false detection/true normal traffic)를 나타낸다. 이 결과에서 Naive Bayes가 가장 성능이 좋지 않은 것을 확인할 수 있고, 제안

되는 CNN 기반 모델이 가장 성능이 좋은 것을 확인할 수 있다. 특히 CNN 기반 모델은 92%이상의 Accuracy를 달성하면서 2%이하의 False positive를 가지는 것을 확인할 수 있었다.

또한, CNN 기반 네트워크 트래픽 분류 기법은 네트워크 트래픽의 특징 벡터를 이미지로 바꾸는 매핑 방법에 무관하게 일정하게 높은 성능을 유지하는 것을 확인할 수 있었다.

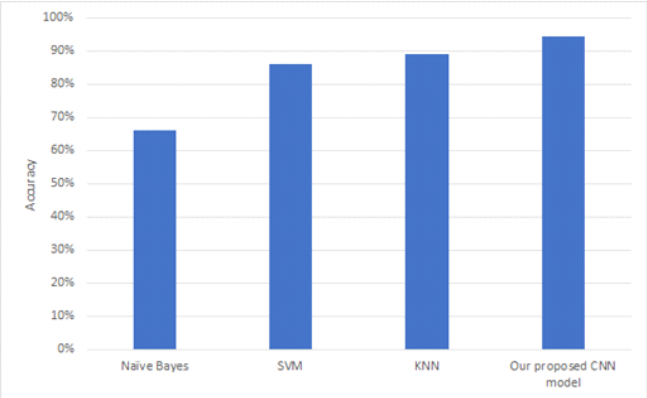


그림 3. 네트워크 트래픽 분류기법 별 Accuracy

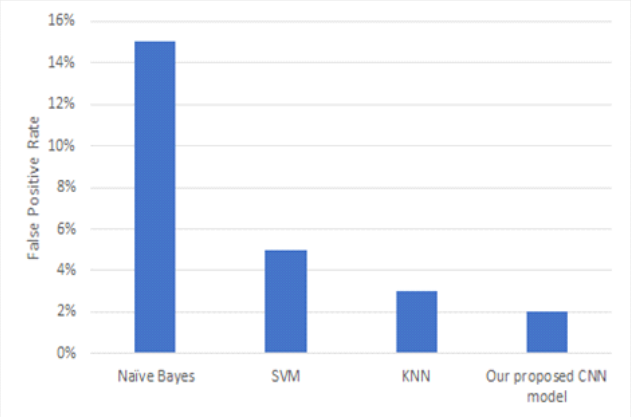


그림 4.네트워크 트래픽 분류기법 별 False Positive

V. 결론

본 논문에서는 유해 네트워크 트래픽을 탐지하기 위한 컨볼루션 신경망(CNN) 기반 네트워크 트래픽 분류 기법을 제안하였다. 제안하는 기법은 네트워크 트래픽의 특징 벡터를 이미지로 변환하고 이를 CNN을 적용하여 분류기 학습에 이용함으로써, DoS 공격 네트워크 트래픽과 일반 네트워크 트래픽을 성공적으로 분류한다. 특히, 특징 벡터의 이미지 변환 시 사용되는 매핑 방법이 분류기 성능에 영향이 거의 없음을 확인하였다. 즉, 네트워크 분류를 위해 원하는 특징벡터를 보다 유연하게 활용할 수 있다는 점을 확인하였다.

제안된 기법은 KDD 1999 데이터셋에서 제공되는 트래픽 특징 벡터를 기반으로 분류 모델을 학습하였는데, 현재 제안된 기법을 적용하여 실제 네트워크 시스템에서 실시간으로 해당 모델을 학습시키는 방법에 대한 연구를 진행 중이다.

ACKNOWLEDGMENT

이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2017RIA2B4012559)). 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학 ICT 연구센터 지원사업의 연구결과로 수행되었음 (IITP-2019-2016-0-00314).

참 고 문 헌

[1] 최진태, 뉘엔 신 응억, 김경백, “유해 네트워크 트래픽 탐지를 위한 트래픽 분류 기법 성능 비교”, 2017년도 한국인터넷정보학회 추계학술대회 논문집 제 19권 2호, 2017

[2] Jintae Choi, Sinh-Ngoc Nguyen, Jeongnyeo Kim, Guee-Sang Lee, Kyungbaek Kim, “Performance Comparison of Traffic Classification Techniques for Detecting Malicious Network Traffic”, In the proceedings of SMA 2017 conference, December 2017.

[3] “KDD Cup 1999 Data.” [Online]. Available: http://kdd.ics.uci.edu/database/kddcup99/kddcup99.html, November 2017.

[4] “Waikato environment for knowledge analysis (weka) version 3.8.1.” [Online] Available : http://www.cs.waikato.ac.nz/ml/weka, October 2017.

활성 사용자 지표 기반 에듀롬 인증로그 분석

장민석, 조부승*
한국과학기술정보연구원
{msjang, bscho}@kisti.re.kr

Log Analysis of eduroam Authentication based on Active User Metrics

Minseok Jang, Buseung Cho

요 약

에듀롬(eduroam)은 전세계 연구교육 활성화를 위한 비영리 Wi-Fi 인허가 로밍서비스이다. 에듀롬은 101 개국 6000 여개 연구소, 대학 등의 Wi-Fi 접속 인허가 인프라를 연계하며, 사용자는 소속기관의 인증정보와 Wi-Fi 접속설정으로 전세계 eduroam SSID 를 편리하게 이용한다. 국제교류확대와 공공와이파이 정책으로 에듀롬에 대한 관심이 증가하고 있으며, 이해하기 쉽고 활용하기 용이한 통계를 내는 것이 중요하다. 대다수의 에듀롬 사용현황 통계는 RADIUS 서버의 인증 건수에 기반한다. 이는 중복이 많고 직관적이 않으며, 다른 IT 서비스와 비교가 불가능하여 활용하기 어려운 지표이다. 본 논문은 중복이 많고, 사용자가 직관적으로 이해하기 힘든 지표를 대체하고자 일간 활성 사용자(DAU)와 월간 활성 사용자(MAU) 지표에 기반한 통계를 제안한다. 또한 유의미한 통계를 도출하기 위해 국내 에듀롬 인프라 및 정책에 관한 발전방향을 제시한다.

I. 서 론

에듀롬(eduroam)은 전세계 연구교육 활성화를 위한 비영리 Wi-Fi 인증 로밍서비스이다. 101 개국 6000 여개 연구소, 대학 등의 협력을 통해 기관과 국가를 뛰어넘는 연구교육환경 조성에 이바지하고 있다. 국제공동연구와 국제교환학생 등 국제교류가 늘어남에 따라 연구자와 학생들이 점차 에듀롬에 관심을 보이고 있으며, 최근 공공와이파이 확대를 위한 정부차원의 관심도 커지고 있다.

대부분의 에듀롬 사용현황 통계는 RADIUS 서버의 인증 건수에 기반하고 있다. 2002 년 서비스를 시작한 이래 2016 년 5 월, 로밍 인증 건수가 10 억 회를 돌파하였다는 기사가 있지만[1], 이는 사용자와 정책입안자의 피부에는 와 닿지 않는다. 얼마나 많이 사용하는지 알기 어려우며, 다른 서비스의 사용량과 비교하기도 힘들다.

또한 RADIUS 인증 건수는 중복이 많다. UDP 에 기반한 RADIUS 패킷의 전송손실을 감안하여 몇몇 관리자들은 RADIUS 서버에서 중복된 패킷을 전송하도록 설정한다. 무선단말은 배터리 절약을 위해 슬립모드 시 무선모듈을 끄기도 한다. 사용자의 이동으로 BSS(Basic Service Set)가 변경되는 경우, Fast Roaming 을 지원하지 않는 무선 인프라는 RADIUS 인증을 다시 수행한다.

본 논문은 중복이 많고, 사용자가 직관적으로 이해하기 힘든 지표를 대체하고자 일간 활성 사용자(DAU)와 월간 활성 사용자(MAU)에 기반한 통계를 제안한다. 또한 유의미한 통계를 도출하기 위해 국내 에듀롬 인프라 및 정책에 관한 발전방향을 제시한다.

II. 에듀롬 소개 및 국내외 에듀롬 현황

에듀롬은 전세계 주요 연구 및 교육기관(연구소, 대학, 학회장, 지자체 등)들의 협약을 통해 Wi-Fi 접속 인허가 인프라를 연계한 것에 기반한다. 사용자는 에듀롬 협약기관에서 별도의 Wi-Fi 사용신청과 계정발급 없이 소속기관에서 Wi-Fi 접속에 사용하는 아이디, 비밀번호, X.509 클라이언트 인증서 등의 인증정보(credential)와 접속설정으로 전세계의 eduroam SSID 를 편리하게 이용한다. 사용자는 인증정보(ID/PW)와 함께 eduroam SSID 에 접속허가를 요청하면, 로밍기반시설을 통해 방문객 소속 기관의 사용자 인증서버에 질의되고, 접속허가를 받아 네트워크를 이용한다(그림 1).

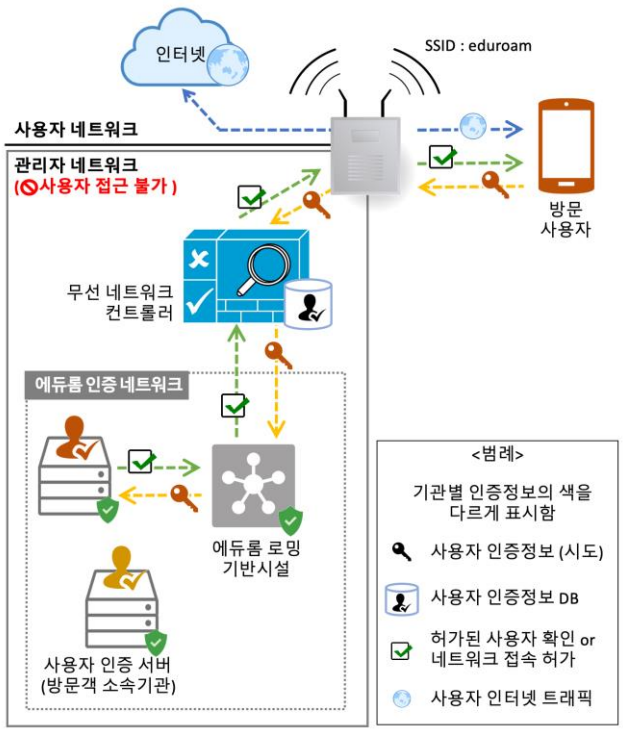


그림 1. 에듀롬 인증 절차

에듀롬은 2003 년 유럽연합(EU)의 연구망인 TERENA (현 GEANT) 주관으로 연구 및 교육 커뮤니티 활성화를 위해 개발되었으며, 현재 미국에서는 520 여개 연구소 및 대학이 참여하고 있다[2]. 국내에서는 2012 년 한국과학기술정보연구원이 운영하는 국가과학기술연구망

(KREONET)이 GeGC(Global eduroam Governance Committee)와의 MOU 를 통해 한국내 대표운영기관(NRO: National Roaming Operator)로 지정되어 한국 내 에듀롬 서비스(eduroam KR) 를 운영하고 있다. 2015 년 국가과학기술연구망은 국공립대학 정보기관협의회와의 협약을 통해 K-에듀롬 참여기관을 글로벌 에듀롬에 연동하였다(그림 2). 현재 국내 69 개 기관 22,000 여개의 AP 를 통해 에듀롬 KR 서비스를 제공하고 있다.

그림 2. 국내외 에듀롬 연동 구조

III. 에듀롬 로그의 중복과 활성 사용자 지표

국내외 다수의 연구교육기관이 협력하여 운영하는 에듀롬은 경계를 뛰어넘는 연구교육환경 조성에 이바지하고 있다. 국제교류가 늘어남에 따라 연구자, 학생, 정책입안자 등 각계 각층이 관심을 보이고 있으며, 이해하기 쉽고 활용하기 용이한 통계를 내는 것이 중요하다.

대다수의 에듀롬 사용현황 통계는 RADIUS 서버의 인증 건수에 기반한다[3]. 인증 건수 만으로는 에듀롬이 얼마나 많이 사용되는지 알기 어려우며, 다른 서비스의 사용량과 비교하기도 힘들다. 또한 인증 건수는 중복이 많은 데이터이다. (그림 3)은 고등과학원 사용자가 해외에서 국내로 인증을 요청한 로그이며, 20 분 동안 15 회의 인증요청이 기록되었다. UDP 기반 RADIUS 인증요청 재전송, FastRoaming 미지원 인프라, 무선 절전모드 등 다양한 요인에 의해 중복이 발생한다.

```
2017-01-01 00:00:46 Auth: Login OK: [n****21@kias.re.kr] (from client eduroam_global)
2017-01-01 00:01:13 Auth: Login OK: [n****21@kias.re.kr] (from client eduroam_global)
2017-01-01 00:05:25 Auth: Login OK: [n****21@kias.re.kr] (from client eduroam_global)
2017-01-01 00:06:34 Auth: Login OK: [n****21@kias.re.kr] (from client eduroam_global)
2017-01-01 00:06:58 Auth: Login OK: [n****21@kias.re.kr] (from client eduroam_global)
2017-01-01 00:07:03 Auth: Login OK: [n****21@kias.re.kr] (from client eduroam_global)
2017-01-01 00:07:05 Auth: Login OK: [n****21@kias.re.kr] (from client eduroam_global)
2017-01-01 00:07:47 Auth: Login OK: [n****21@kias.re.kr] (from client eduroam_global)
2017-01-01 00:07:57 Auth: Login OK: [n****21@kias.re.kr] (from client eduroam_global)
2017-01-01 00:08:14 Auth: Login OK: [n****21@kias.re.kr] (from client eduroam_global)
2017-01-01 00:09:11 Auth: Login OK: [n****21@kias.re.kr] (from client eduroam_global)
2017-01-01 00:09:54 Auth: Login OK: [n****21@kias.re.kr] (from client eduroam_global)
2017-01-01 00:11:31 Auth: Login OK: [n****21@kias.re.kr] (from client eduroam_global)
2017-01-01 00:19:23 Auth: Login OK: [n****21@kias.re.kr] (from client eduroam_global)
2017-01-01 00:19:55 Auth: Login OK: [n****21@kias.re.kr] (from client eduroam_global)
```

그림 3. 중복된 인증로그

중복을 제외하고 직관적으로 이해하기 쉬운 지표로는 활성 사용자(Active User)가 있다[4-5]. 활성사용자는 기간에 따라 일간 활성 사용자(DAU, Daily Active User), 주간 활성 사용자(WAU, Weekly Active User), 월간 활성 사용자(MAU, Monthly Active User) 등으로 나뉘며, DAU 는 해당 서비스를 하루에 몇 번을 사용하든지 간에 1 회로 집계한다. 활성 사용자는 사용자 참여(User Engagement)를 측정하는 가장 기본적인 단위이며 광고, 마케팅 영역에서 사용되는 지표이다. 광고주가 시장의 크기를 알고 광고를 집행하는데 참고하거나, 스마트폰 게임에서 예상되는 광고 수입 계산할 수 있다. 활성 사용자의 추이를 보고 사전에 인프라를 증축하거나 외부의 투자를 받기도 한다.

또다른 중요한 지표로는 사용자가 서비스를 얼마나 자주 사용하는지 확인할 수 있는 지표인 고착도(stickness)가 있다. 고착도는 지난 한달 간 특정 서비스를 사용한 사람이 오늘도 사용할 확률로 일간 활성 이용자 수를 이전 30 일간의 월간 활성 이용자 수로 나누어 구한다[6]. Google Analytics[7]와 같은 여러 도구들이 웹사이트나 게임에 단순한 코드 삽입만으로 이러한 지표들을 손쉽게 수집 및 계산하는데 도움을 주고 있다.

IV. 에듀롬 KR 의 NRO 인프라 구성 및 데이터 전처리

국가과학기술연구망은 2012 년도에 어플라이언스 형태의 RADIUS 장비 2 개를 Active - Standby 로 구성하여 에듀롬 KR 의 NRO 인프라로 구성하였다. 해당 장비는 FreeRadius 에 기반한 SW 를 탑재하고 있으며, 각기 별도의 내장 DB 를 가지고 있으며 WebUI 이외의 조회는 불가능하다. 인증로그는 로그는 rotate 되어 년/월/일 폴더에 쌓이며, 이를 년/월/일 단위로 tgz 로 묶어 다운로드 하는 것만 가능하며, 로그의 포멧은 변경하지 못한다. 어플라이언스 제품 중에서도 상당히 제약이 많은 장비이다.

우선 텍스트 기반 로그를 파이썬으로 전처리 한다. 로그는 예외사항이 많으며, 단순한 정규식(Regular Expression)으로 처리할 수 없어 전처리 과정이 필수적이다. 1) 디버그 로그와 인증 로그가 섞여있어, 디버그 로그를 제외한다. 2) 문자열 인코딩을 강제로 변환(truncate)한다. 해당 장비의 로그파일의 문자열은 한글 완성형 인코딩(EUC-KR) 이지만, 해외에서 사용자 ID 나 원격 RADIUS 서버 정보 등이 다른 문자열 세트로 표현되어 로그에 기록되고 깨져서 보이게 된다. 이를 강제로 유사한 문자로 변경한다. 3) 개행문자, 화면에 보이지 않는 특수문자, 코드 등을 치환한다. 파서에 에러를 유발하는 문자를 치환한다. ID 필드에 특수문자, 개행문자를 넣거나 코드를 삽입하는 등 프로그램의 오동작을 유발하는 포멧 스트링 공격, SQL 인젝션 공격 등을 시도한 정황이 로그에 남아 있다. 예를 들어 개행문자는 1 줄의 로그를 인증시도 1 건으로 기대하는 파서에 에러를 유발하므로 이를 치환하여 1 줄로 만든다. 4) 맥 주소를 표기하는 방법도 Wi-Fi AP 및 컨트롤러에 따라 4 가지 이상으로 다양하기에 이를 하나의 형식으로 통일한다. AB:0C, ab:0c, ab-0b, ab0c 등 16 진수를 다양하게 표현 가능하며 이를 첫 번째 방식으로 통일한다. 그 외 발생하는 에러를 수정하며 전처리기와 파서를 완성한다.

V. 활성 사용자 지표 기반 에듀롬 인증로그 분석

전처리 된 로그는 파이썬 정규식 등을 활용한 로그분석기로 파싱하여 csv 파일로 변환하며, 이를 Pandas 로 로딩하여 로그를 분석한다. 우선 datetime

컬럼으로부터 month(월), weeknum(주차), date(일), day(요일)의 4 개 컬럼을 생성한다. 성공한 인증에 대해 id 와 date 컬럼을 함께 select 한 뒤 겹치는 것을 제외 - drop_duplicate() 하여 일별 활동 사용자 목록을 얻는다. 이를 date 컬럼에 대해 그룹화 하고 count()하여 일별 활동 사용자 수 (DAU)를 얻는다. 유사한 방법으로 주차별(WAU), 일별(DAU), 요일별 사용자 수를 구한다. 만약 SQL 에 로그 데이터가 들어있다면, DISTINCT, GROUPBY, COUNT 등과 서브쿼리를 활용한 쿼리를 만들어 얻을 수 있다.

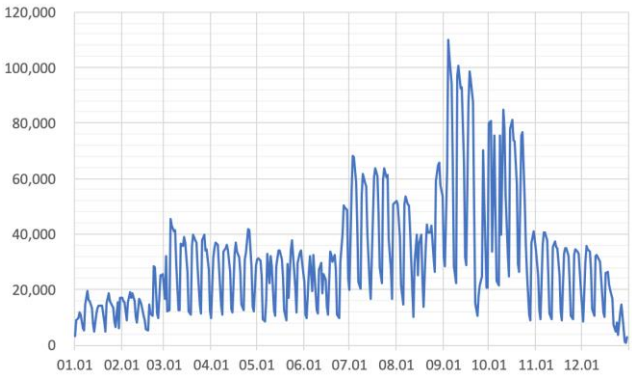


그림 4. 일간 인증 건수 (2018 년)

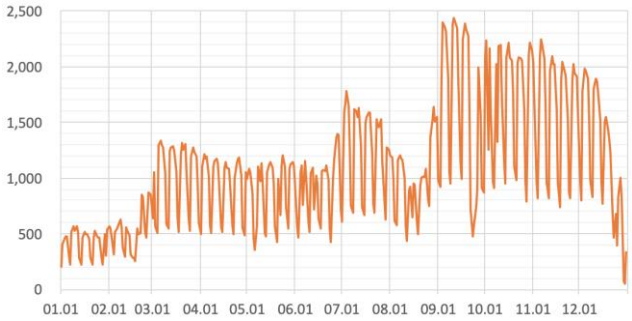


그림 5. 일간 활성 사용자 수 (2018 년)

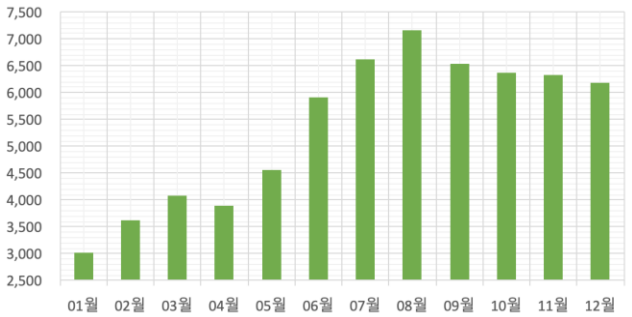


그림 6. 월간 활성 사용자 수 (2018 년)

위와 같은 과정을 통하여 에듀롬 KR 의 일간 및 월간 활성 사용자 수를 구하였으며 여러가지 사실을 확인할 수 있다. 1) 일간 인증 건수와 일간 활성 사용자 수 모두 1 주일을 주기로 사용량이 변하는 것을 확인할 수 있다. 주중에 사용하는 사람이 많은 것을 관찰할 수 있다. 2) 일간 인증 건수와 일간 활성 사용자 수를 비교하면 8~45 배 차이가 난다. 12 월 31 일에 최소, 9 월 4 일에 최대이다. 여름방학 기간에는 전 기간에 걸쳐 30 배 이상 차이가 나는데, 이는 이동이 많아 인증요청이 많이 발생하는 것으로 추측된다. 3) 10 월과 대비하여 11~12 월에는 인증 건수는 확연히 줄어든 반면 활성 사용자 수는 유지되는 것을 확인할 수 있다. 이는 RADIUS 서버의 재전송 문제 등이 확실히 개선되었거나, 사용자들의 이동이 줄어 들어 인증 요청이 감소하였을 가능성이 있다. 4) 7~9 월에 일간 활성 사용자 수와 월간 활성 사용자 수를 비교하면, 8 월에는 일간 활성 사용자 수는 9 월 보다 적지만, 월간 활성 이용자 수는 반대로 7~8 월이 9 월보다 높은 것을 확인할 수 있다. 5) 일간 활성자 수를 이전 30 일의 월간 활성 이용자 수로 나눈 고착도를 계산해 보면, 10~36%가 나온다. 에듀롬의 실질적인 사용자를 월간 활성 사용자로 한정하면, 사용자는 1 달에 3~10 일을 에듀롬을 사용하는 것을 확인할 수 있다. 또한 Facebook 의 고착도는 50% 정도이며, 성공한 게임의 경우 초기 고착도는 20%인 것[4]을 감안하면 에듀롬은 성공적으로 운영되고 있는 서비스라 할 수 있다.

VI. 국내 에듀롬 인프라 및 정책 개선방향

본 논문에서 지금까지 기술한 사항은 국가과학기술연구망이 운영하는 RADIUS 서버의 인증로그만을 분석한 것이다. 본 서버는 한국과 외국의 관문으로, 주로 한국내 외국인이 본국에 인증을 요청하거나 외국 내 한국인 사용자가 한국의 소속기관에 인증을 요청할 때 인증요청이 경유되는 지점으로, 주로 이에 대한 인증 요청만 분석되었다.

국내 사용자의 다수는 고등교육 기관에 속한 대학생이며, 학점교류 등을 위해 타 대학의 캠퍼스에 방문하는 일이 많을 것으로 예상된다. 국내 대부분의 대학은 (그림 2)에서 언급한 것과 같이 고등교육부분 RO 와 연동되어 있으며, 대학간 RADIUS 인증은 집계가 되지 않는다. 때문에 국내 현황에 대해 심도 있는 분석을 위해서는 국가과학기술연구망이 운영하는 NRO 의 로그와 국공립대학 정보기관협의회가 운영하는 RO 의 로그를 서로 공유하며 통합분석하여야 하며, 이를 위한 정책 수립과 인프라 투자가 필요하다.

VII. 결론

본 논문에서는 활성 사용자 지표에 기반하여 에듀롬 인증로그를 분석하였다. 중복과 예외가 많은 RADIUS 로그를 전처리 하고 일간 및 월간 활성 사용자 수를 구하였다. 에듀롬 KR 은 일일 최대 2500 여명이 이용하는 서비스로 7100 여명의 활성 사용자가 이용하고 있으며, 고착도는 10~36%로 성공적으로 운영되고 있는 서비스임을 확인하였다. 심도 있는 국내 에듀롬 사용자 분석을 위해서는 국내 에듀롬 참여기관의 로그 공유와 통합분석이 필요하며, 이를 위한 정책 수립과 인프라 투자가 필요하다.

ACKNOWLEDGMENT

본 연구는 과학기술정보통신부 및 한국과학기술정보연구원의 주요사업의 일환으로 수행하였음. [과제명: 글로벌 협업연구 지원 국가 과학기술연구망 구축 및 서비스]

참 고 문 헌

[1] "eduroam celebrates one billion roaming authentications", May 2016, (https://www.eduroam.org/happy-1000000000th).

[2] "List of US Institutions - eduroam US", (https://www.eduroam.us/institutions_list).

[3] "eduroam Statistics (F-Ticks) per country", (https://monitor.eduroam.org/f_ticks_country.php).

[4] Lalmas, Mounia, Hong, Liangjie. "Tutorial on Metrics of User Engagement: Applications to News, Search and E-Commerce", Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 781-782, 2018

[5] Y. M. Kassa, R. Cuevas and Á. Cuevas, "A Large-Scale Analysis of Facebook's User-Base and User Engagement Growth," in IEEE Access, vol. 6, pp. 78881-78891, 2018.

[6] "사용자 앱 의존도를 파악하는 5 가지 지표", (https://brunch.co.kr/@userhabit/19)

[7] "Active Users - Google Analytics", (https://support.google.com/analytics/answer/6171863).

확장성 있는 이벤트 알림 봇 설계 및 구현

문현수, 이영석
충남대학교

munhyunsu@cnu.ac.kr, lee@cnu.ac.kr

Design and implementation of scalable event notification bot

Hyunsu Mun, Youngseok Lee
Chungnam National University

요 약

카카오톡, *Telegram*, *Twitter* 등 다양한 SNS 가 제공하는 유용한 봇 서비스는 고객 응대를 위한 QnA 서비스 봇, 결제 정보 알림 봇 등 기업들이 사용자와 소통할 수 있는 새로운 창구로 떠오르고 있으며 동시에 개인들도 사용하기 쉬운 봇 API 를 이용해 간단한 질의 응답 봇, 정보 봇을 만들어 활용하고 있다. 하지만 QnA 봇, 정보 알림 봇이 활용할 데이터를 자체적으로 생산할 수 있거나 고정되어 있지 않으면 데이터 수집부터 구현해야 하고, 이에 따라 적절한 시간(예. 행사 시작 6 시간 전)에 알림 기능을 설정하기 어렵다. 특히 알림 서비스를 제공할 이벤트가 비정기적으로 열릴 경우 시간 관리 모듈 추가로 프로그램 구조가 복잡해져 유지보수가 어려워진다. 데이터 수집과 알림 시간 관리 문제는 클래스 모듈화된 수집, 알림 기능을 하나의 프로세스가 불러와 동작하여 해결할 수 있다. 본 논문에서는 데이터를 수집하고 정해진 시간에 포스팅하는 봇 설계 및 구현 방법을 제안하고 웹 사이트에 공개되는 비정기 행사 정보를 수집, 특정 시간(행사 시작 6 시간, 행사 시작, 행사 끝)에 포스팅을 하는 봇 구현 결과를 보인다. 제안한 소프트웨어 설계 및 구현 방법을 통하여 정보 수집 및 포스팅뿐만 아니라 기존에 서비스되고 있는 웹을 수정하지 않고도 쉽게 봇 서비스로 확장할 수 있다. 또한 우리는 Twitter 봇으로 비정기 이벤트 알림 서비스를 제작하여 서비스한 1 년간의 운용 이슈도 함께 보인다.

Ⅰ. 서 론

기존 *Cyworld* [1], 블로그 등으로 대표되는 SNS 서비스는 이동통신과 모바일 디바이스의 발전으로 좀 더 개인화되고 생활에 밀접한 *카카오톡* [2], *Telegram* [3], *Twitter*[4], *Instagram*[5] 등으로 성장하였다. 이러한 SNS 서비스는 개인 사용자들이 활용할 채팅, 마이크로블로깅 서비스뿐 아니라 기업과 같은 단체가 고객을 대상으로 정보를 제공할 봇 서비스를 지원한다 [1]. 기업 계정 관리 직원이 직접 정보를 포스팅하는 봇, QnA 봇, 그리고 결제 정보 봇이 기업과 사용자 간의 새로운 소통 창구로 활용되고 있다. 특히 2024 년 21 억 달러 규모로 성장할 것으로 예측되는 채팅 봇은 QnA 와 고객 응대 서비스를 제공하고 있으며 다양한 서비스의 새로운 인터페이스로 떠오르고 있다.

기업 봇과 동시에 봇 API 를 이용해 쉽게 개발할 수 있는 장점으로 인하여 개인이 직접 봇을 제작하여 활용하는 사례가 많아지고 있다 [2]. 이러한 봇 서비스가 사용자에게 잘 활용되기 위해서는 제공할 데이터에 쉽게 접근할 수 있어야 한다. 따라서 데이터를 생성하기 위하여 기존의 정보를 활용하는 연구가 진행되고 있다 [3, 4]. SNS 서비스가 제공하는 봇 API 를 활용하면 쉽게 봇을 제작하고 운용할 수 있기 때문에 사용자 의도에 맞게 잘 가공하고, 적절한 시간에 제공하는 것이 필수적이다. 따라서 QnA, 정보 알림 봇은 활용할 데이터를 자체적으로 생산할 수 있거나 응답 정보를 고정시켜두어 서비스한다. 만약 변동되는 데이터를 서비스해야할 때에는 봇이 활용할 데이터를 구출할 전용 모듈을 추가한다.

봇은 사용자에게 적절한 데이터를 제공하는 것이 필수적임에도 불구하고 웹에서 접근가능한 데이터를 봇에서는 쉽게 활용하기 어렵다. 1989 년 Tim Bernes-Lee 가 World Wide Web 을 개발한 이래로 웹에는 다양한 정보가 쌓여가고 있다. 이러한 정보들은 HTML, CSS, Javascript 등을 통하여 지속적으로 생성되고 업데이트되어 웹에 공개되고 있다. 이러한 정보를 봇이 제공하기 위해서는 수집기와 봇 API 를 연결해야하기 때문에 프로그램 구조가 복잡해진다. 특히, 비정기적인 이벤트 정보를 제공해야할 경우 시간 관리 모듈이 스케쥴링도 해주어야하여 개인이 빠르게 구현하여 활용하기에는 프로그램이 거대해진다.

본 논문에서는 데이터를 수집하고 정해진 시간에 포스팅하는 봇 설계 및 구현 방법을 제안하고 웹 사이트에 공개되는 비정기 행사 정보를 수집, 특정

[1] http://www.cyworld.com/
[2] https://www.kakaocorp.com/service/KakaoTalk
[3] https://telegram.org/
[4] https://twitter.com/
[5] https://www.instagram.com/

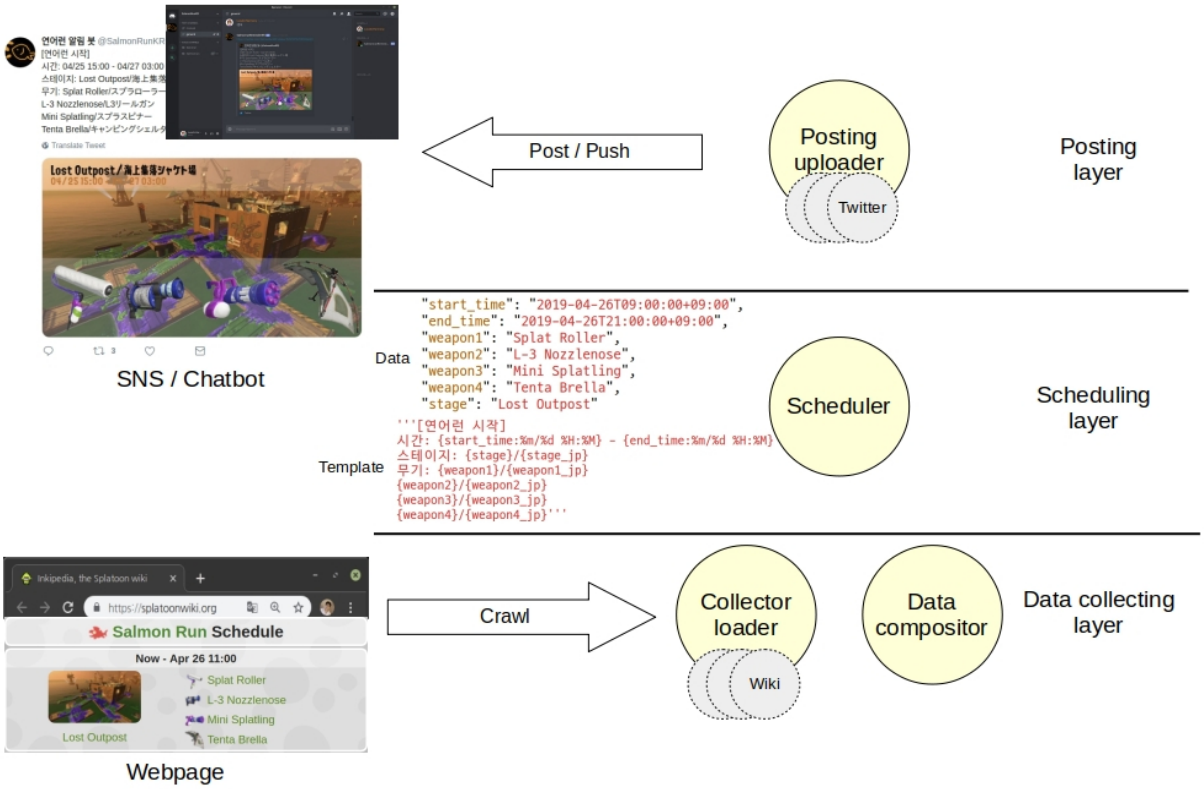


그림 1. 확장성 있는 이벤트 알림 봇 구조.

시간(행사 시작 6 시간, 행사 시작, 행사 끝)에 포스팅을 하는 봇 구현 결과를 보인다. 개인이 빠르게 구현하기 어려울 정도로 복잡해지는 데이터 수집과 알림 시간 관리 문제는 인터페이스가 맞춰진 클래스로 수집, 알림 기능을 하나의 프로세스가 불러와 동작시키는 것으로 해결할 수 있다. 서비스 개발하고자 할 때 수집 클래스와 포스팅 클래스를 구현하는 것으로 원하는 정보를 의도한 시점에 자동으로 포스팅되게 할 수 있다. 이러한 설계를 통하여 기존에 서비스되고 있는 World Wide Web 과 같은 데이터 소스를 그대로 이용하여 봇 기능을 확장할 수 있음에 기여한다. 또한 우리는 이러한 설계를 통하여 Wiki 에 자동으로 업데이트되는 정보를 수집하고 특정 시간에 맞추어 Twitter 로 정보를 포스팅하는 봇을 1 년간 운영한 결과를 보인다.

Ⅱ. 확장성 있는 이벤트 알림 봇

Ⅱ.1. 이벤트 알림 봇 구성 요소 및 구조

확장성 있는 이벤트 알림 봇(그림 1)은 Collecting, Scheduling, Posting 레이어로 구성된다. Collecting, Posting 레이어는 새로운 데이터 소스나 포스팅 서비스 API 를 연결하기 위하여 사용되며, Scheduling 레이어에서는 포스팅할 시간을 계산한다. 각 레이어는 독립적으로 동작하기 때문에 새로운 모듈 추가/제거가 다른 모듈에 영향을 주지 않는다. 레이어 독립성을 통하여 본 연구에서 제안하는 이벤트 알림 봇은 확장성을 가진다.

Collector loader 는 구현된 Collector subclass 모듈을 동적으로 로드하고 객체를 생성 및 메소드를 실행하는 역할을 한다. 코드 1 은 Collector loader 의 주요 코드로 구현되어 있는 데이터 수집기를 동적으로 모두 로드, 실행한다. 각 데이터 수집기는 포스팅할 내용과 서식(print format), 그리고 포스팅 시간을 출력한다. **Data compositor** 는 출력된 데이터를 JSON 포맷으로 저장한다(그림 1).

Scheduler 는 저장된 JSON 파일을 읽어와 포스팅 시간이 되었는지 확인하고 Posting uploader 를 수행할지 결정한다. 코드 2 는 Scheduler 의 주요 코드로 읽어온 포스팅 시간과 현재 시간을 비교한다. Scheduler 는 crontab 으로 주기적으로 실행되기 때문에 crontab 주기에 맞추어서 threshold(thres) 값을 정해줄 수 있다. 만약

코드 1. Collector 서브 클래스 동적 로드 및 실행.

```
# Load all of collector subclass
for _, name, _ in pkgutil.iter_modules([FLAGS.coll_path]):
  imported = importlib.import_module('colls'+'.'+name, name)
  class_name = name.split('_')[-1]
  class_object = getattr(imported, class_name)
  collectors.append(class_object())

# Scrap data from web using get_data functionfor collector in collectors:
  collector.get_data()
```

코드 2. Scheduler 포스팅 시간 확인 (시작시간 전후 30 분내 포스팅)

```
# The function for check posting schedule
def get_schedule(schedule, thres, now=datetime.datetime.now(TIMEZONE)):
  start_time = ''.join(schedule['start_time'].rsplit(':', 1))
  start_time = datetime.datetime.strptime(start_time, ISO8601KST)
  start_sec = (start_time - now).total_seconds()
  if (start_sec < thres) and (start_sec > -thres):
    return schedule
  return None
```

그림 2. 비정기 이벤트 Twitter 봇 포스팅 결과.

포스팅해야 할 시간이 되었다면 Scheduler 는 직접 Posting uploader 를 실행하여 데이터를 전달한다.

Posting uploader 는 포스팅해야 할 JSON 데이터를 입력 받아 SNS API 를 이용하여 내용을 업로드한다. 실행 타이밍은 Scheduler 가 조정해주기 때문에 Posting uploader 는 입력으로 들어온 JSON 데이터를 서식에 맞추어 업로드하는 기능만 담당한다. Scheduler 가 해당 SNS API 를 이용하여 구현된 Posting uploader 서브 클래스를 직접 호출할 수 없기 때문에 Posting uploader 는 직접 알맞은 서브 클래스를 찾아 메소드를 실행해야 한다.

Ⅱ.2. 비정기 이벤트 Twitter 봇 구현

우리는 제안한 봇 구조가 동작함을 보이기 위하여 2018 년 05 월부터 Inkipedia(게임 위키)에서 데이터를 수집하고 특정 시간에 Twitter 로 포스팅하는 봇을 구현하고 운영하였다. 사용자 편의성을 위하여 포스팅 내용에 맞는 이미지도 자동으로 제작하는 기능을 추가하였다.

그림 2 는 구현한 트위터 봇이 비정기 이벤트의 시작 6 시간 전, 시작, 종료 1 시간 전, 종료 시간마다 자동으로 포스팅한 결과다. 포스팅하는 정보는 기존에 서비스되고 있는 Wiki 에서 수집하여 활용하였기 때문에 새로운 정보원을 만들지 않고 기존에 운영되던 서비스 정보를 그대로 활용할 수 있었다. 또한 이미지를 자동으로 생성하는 기능을 추가하여 제안한 구조가 원하는 기능을 강화하는 것도 가능함을 보였다.

그림 1 의 Posting layer 에 여러 클래스를 두는 것으로 제공 서비스를 확장할 수 있다. 그림 3 은 Inkipedia 수집기가 저장한 데이터를 그대로 이용하여, Discord [6] 채팅 봇 서비스를 운영하는 사진이다. crontab 을 통하여 주기적으로 최신 데이터가 수집되기 때문에 비정기 이벤트의 가장 최신 정보를 제공할 수 있다. 따라서 우리가 제안한 봇 구조는 데이터 수집뿐 아니라 서비스 면에서도 확장성을 가진다.

[6] https://discordapp.com/

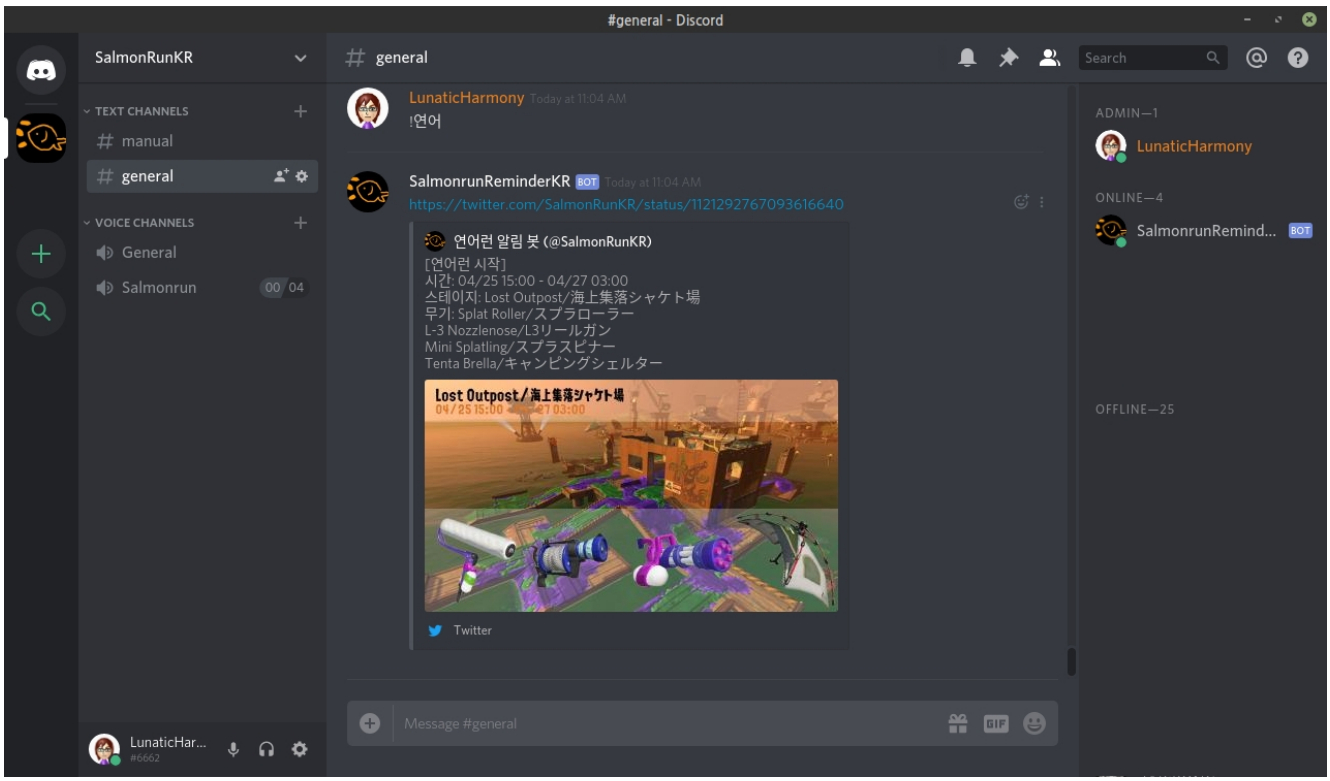


그림 3. Twitter 봇에서 활용하는 정보를 그대로 이용하는 Discord 채팅 봇.

우리가 제안한 봇 구조를 통해 기존에 운영되는 서비스를 이용하여 포스팅 봇, 채팅 봇을 운영하는 것은 자원을 재활용하고 확장이 쉽다는 장점이 있다. 하지만, 데이터가 기존 서비스에 의존성을 가지게 되어 운영상에 문제가 생기기도 한다. 2018 년 05 월부터 1 년간 Inkipedia 정보를 이용하여 서비스하는 동안 데이터 원에서 시작된 문제가 생겼다(표 1). 특히 웹 서비스에서 데이터를 수집하는 것이기 때문에 HTML 레이아웃에 민감하다. HTML 레이아웃이 변경되면 수집기의 HTML 문서 파서도 이에 맞추어 업데이트하여 해결할 수 있다.

표 1. Twitter 봇을 1 년간 운영하며 발생한 Inkipedia (데이터 출처) 로 인한 서비스 문제 및 횟수.

문제	횟수
서비스 레이아웃 변경(HTML)	4
정보 업데이트 멈춤	1
잘못된 정보 업데이트	1

Posting uploader 는 SNS API 를 사용하는 것이기 때문에 실시간 알림에는 적합하지 않다. 주기적으로 데이터를 수집해두고 갖추어 요청을 받자마자 포스팅하더라도 SNS 서비스의 상황에 따라서 개인에게 푸시 알림이 도착하는 시간이 다르다. crontab 수행 주기를 빠르게 하여 동작 시키더라도 사용하는 SNS 사양에 따라서 동작 시간의 최소값이 결정된다. API 활용을 위한 사용자 인증도 포함하는 Twitter 는 Posting uploader 동작 후 SNS 에서 포스팅을 확인할 수 있을 때까지 최소 3 초 이상이 걸린다.

Ⅲ. 결론

본 논문에서는 기존에 서비스되고 있는 정보를 수집하여 비정기 이벤트에 대한 정보를 알려주는 봇 설계 및 구현 방법을 제시하고 그 구현 및 운영 결과를 보였다. 웹 서비스 Collector 와 SNS API 를 활용한 Posting uploader, 그리고 Scheduler 를 활용하여 간격에 규칙이 없는 이벤트 알림을 서비스하였다. 2018 년 05 월부터 Twitter 봇 및 Discord 채팅 봇을 운영하며 기존 서비스 의존성으로 인한 문제가 있지만 기존 데이터 재활용 및 서비스 확장의 용이성을 보였다. 이 결과는 기존 서비스를 활용하여 새로운 서비스로 발전시키고, 동시에 확장성을 유지하는 방법에 기여한다.

ACKNOWLEDGMENT

이 논문은 2016 년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (NRF-2016R1D1A1A09916326).

참 고 문 헌

[1] A. Xu, Z. Liu, Y. Guo, V. Sinha, and R. Akkiraju, “A new chatbot for customer service on social media.” In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 3506-3510, ACM, 2017.

[2] 김영욱, 이강호, “챗봇과 마이크로소프트 봇 프레임워크,” 한국컴퓨터정보학고히지, vol. 25, no. 2, pp. 9-15, 2017.

[3] M. Fischer and M. Lam, “From books to bots: Using medical literature to create a chat bot,” in Proceedings of the First Workshop on IoT-enabled Healthcare and Wellness Technologies and Systems, pp. 23-28, ACM, 2016.

[4] 홍금원, 이정훈, 신중휘, 이도길, 임해창, “대화시스템의 로그를 이용한 대화예제의 자동 확중에 관한 연구.”, 한국 HCI 학회학술대회, pp. 257-262, 2017.

M-CORD 모니터링 시스템을 이용한 비정상 상태 탐지 연구

박수현°*, 홍지범*, 유재형†, 홍원기*

*포항공과대학교 컴퓨터공학과
†포항공과대학교 정보통신대학원

{sh.park11, hosewq, styoo, jwkhong}@postech.ac.kr

A Study of Anomaly Detection on the MCORD Monitoring System

Suhyun Park°*, Jibum Hong *, Jae-Hyoung Yoo†, James Won-Ki Hong*

*Department of Computer Science and Engineering, POSTECH
†Graduate School of Information Technology, POSTECH

요 약

Mobile-CORD(M-CORD)는 기존 전화국에서 운용되는 모바일 LTE 네트워크를 클라우드 데이터센터 또는 엣지 클라우드(edge cloud)를 통해 제공하는 플랫폼으로, 이를 통해 다양한 트래픽 상황에 대해 신속하고 유연하게 대응할 수 있다. 하지만 이와 같이 가상 네트워크 기능(Virtual Network Function)을 활용하는 경우 컴퓨팅 자원 관리 및 네트워크 성능에 문제가 발생할 수 있다. 본 논문에서는 M-CORD 모니터링 시스템을 통해 수집되는 컴퓨팅 및 네트워크 자원 사용 데이터를 대상으로 인공지능을 활용하여 LTE EPC 네트워크 구성 요소의 비정상 상태를 탐지하는 시스템을 제안한다. 이를 통해 M-CORD 플랫폼으로 제공되는 LTE EPC 네트워크를 보다 효율적으로 관리 및 운용할 수 있다.

Ⅰ. 서 론

CORD (Central Office Re-Architected as a Data Center)는 기존 전화국에서 운용하는 하드웨어 중심의 통신장비를 소프트웨어화 및 가상화를 통해 서버에서 동작시켜 전화국을 데이터센터와 같이 운용할 수 있는 데이터센터 플랫폼이다. 기존의 전용 하드웨어 통신장비들은 각각의 기능을 갖고 고정적으로 배치되는 반면, CORD 는 다음의 세 가지 기술을 결합하여 네트워크 기능을 유연하게 구성할 수 있다. 첫째, 네트워크 제어 평면(control plane)과 데이터 평면(data plane)을 분리하여 네트워크를 제어 및 관리하는 소프트웨어 정의 네트워킹(Software-Defined Networking, SDN)이다. 둘째, 하드웨어 장비의 네트워크 기능을 범용 서버 상의 가상 머신을 통해 수행하는 네트워크 기능 가상화(Network Function Virtualization, NFV)이다. 마지막으로 소프트웨어기반 솔루션으로 서비스를 탄력적으로 확장할 수 있도록 하는 클라우드 컴퓨팅(Cloud Computing) 기술이다[1]. Mobile-CORD(M-CORD)는 CORD 플랫폼을 통해 모바일 LTE 네트워크를 서비스하기 위해 기존 LTE EPC(Evolved Packet Core) 네트워크의 구성요소를 가상화하여 CORD 플랫폼 내 서버에서 운영하고, 이를 기지국(evolved NodeB, eNB)과 연결하여 모바일 네트워크를 구축한다.

하지만 네트워크의 기능을 가상화하면서 복잡한 가상 네트워크 환경에서 동작하는 VNF 들의 컴퓨팅 자원 할당 및 관리와 네트워크의 성능 보장 및 부하 관리 문제가 존재한다[2]. 이러한 NFV 환경에서의 관리 문제를 해결하기 위해 가상 머신 및 VNF 의 비정상 상태를 탐지할 필요가 있다.

본 논문에서는 M-CORD 플랫폼에서 EPC 네트워크를 구성하는 컨테이너들의 자원 사용 상태를 모니터링 시스템을 통해 수집하고, 머신러닝 기법을 통해 학습하여 실시간으로 EPC 네트워크 구성요소의 비정상적인 상태를 탐지(Anomaly Detection)하는 시스템을 제안한다.

Ⅱ. 관련 연구

수집된 자원 사용량 데이터를 통한 시스템의 비정상 탐지는 어떤 기준으로 비정상 상태를 판단할 것인지에 따라 다양한 방법으로 접근할 수 있다. 예를 들어 CPU 사용률에 따른 비정상 탐지를 하는 경우, 특정 시간 동안의 CPU 사용률 데이터에 대해 평균과 표준편차를 구한 후, 3 시그마 규칙(three-sigma rule)을 적용하여 평균에서 표준편차의 3 배 이상 떨어져 있는 지점을 임계값(threshold)으로 삼아 비정상 상태를 규정할 수 있다[3].

수집되는 데이터는 각 시점에 측정되는 값으로 시계열 데이터이고, 시계열 데이터에 대한 비정상 탐지는 주기적 변화나 경향성에 따른 값의 변화를 배제하기 위한 STL 분석(Seasonal-Trend Decomposition) 방법 등이 이용된다. 이를 활용한 연구[3]에서는 시계열 데이터에 대한 특성을 고려하여 구간의 이동 평균(Simple Moving Averages, SMA)을 이용하여 비정상에 대한 임계값을 정하는 방법이 이용되었다.

적용된 방법의 효과를 평가하기 위해 수집한 데이터를 이용한 VNF 의 정상 및 비정상 상태에 대한 분류(labeling)가 선행되어야 하는데, 각 시점에서 정상/비정상이 분류된 데이터를 구하는 것이 현실적으로 어렵다. 따라서 특정 시점에 비정상 데이터를 인위적으로 발생시켜 그 시점의 데이터를 비정상으로 분류하는 방법이 이용되었다. 이와 유사한 다른 비정상 탐지 연구[4]에서는 각각의 모니터링 데이터에 대한 임계값 대신 여러 특성(feature) 조합의 분포를 엔트로피로 정의하고 비정상 탐지 후, 결과를 임계값 기반의 접근 방식(threshold-based detection)과 비교하여 효율성을 평가하였다.

최근 여러 분야에서 인공지능을 이용한 비정상 상태 탐지 연구도 활발히 진행되고 있다. 선행 연구[5]에서는 비정상 탐지의 문제를 정상/비정상 상태의 분류(classification) 문제로 접근하여 비정상 상태의 초기 징후를 Random Forest 를 이용한 인공지능 기반 모델로 탐지하였다.

Ⅲ. M-CORD 비정상 상태 탐지 시스템

본 논문에서 제안하는 M-CORD 모니터링 시스템을 활용하여 비정상 상태를 탐지하는 시스템의 구조는 그림 1 과 같다.

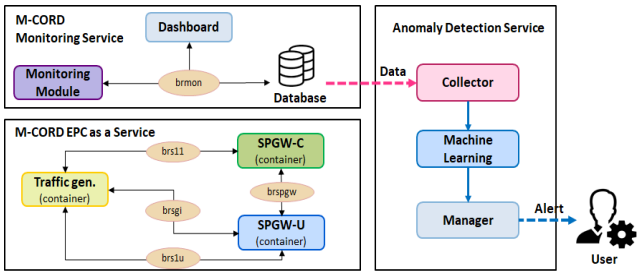


그림 1 M-CORD 비정상 상태 탐지 시스템의 구조

먼저, M-CORD 의 EPC 네트워크는 Serving Gateway(S-GW)와 PDN Gateway(P-GW)를 통합하여 제어 평면과 데이터 평면으로 나눈 SPGW-C 와 SPGW-U, 그리고 이들을 검증하기 위한 eNB 에뮬레이터(Traffic-gen)가 컨테이너(container)로 구성되어 있다. 모니터링 시스템은 각 컨테이너들이 사용하는 컴퓨팅 및 메모리 자원(CPU 및 메모리 사용량, 컨테이너들이 송/수신하는 데이터 트래픽 등)을 모니터링 모듈을 통해 수집하고, 데이터베이스에 저장하여 이를 Web UI(Dashboard)를 통해 보여준다.

비정상 상태 탐지 시스템은 각 EPC 컨테이너들의 모니터링 데이터를 수집하고 학습에 이용할 수 있도록 전처리(preprocessing)하는 과정, 수집된 데이터를 바탕으로 RNN (Recurrent Neural Networks) 알고리즘을 통해 학습시켜 비정상 탐지 모델을 만드는 과정, 그리고 만들어진 모델을 시스템에 적용하여 관리자에게 경보를 생성하는 과정으로 구성된다.

학습 및 검증 데이터를 수집하는 과정에서는 임의의 주기로 결함을 주입(fault injection)하여 비정상 데이터를 발생시키고, 각 시점의 모니터링 데이터를 입력 값으로, 결함 주입 여부를 결과값으로 수집한다. 수집된 데이터는 샘플링 과정을 거쳐 특징을 파악하고, 필요한 경우 각 항목별로 정규화 또는 값을 변환하는 등의 과정으로 데이터 전처리를 수행한다.

모델의 학습 과정을 위해 전처리가 완료된 데이터를 학습 및 검증 데이터로 나눈 뒤, 시계열 데이터를 학습할 때 주로 사용되는 RNN 의 변형인 LSTM(Long Short-Term Memory) 알고리즘을 적용하여 각 시점의 모니터링 데이터인 입력값에 대해 각각 비정상/정상 여부를 결과값으로 내도록 모델링하여 학습을 진행한다. LSTM 알고리즘을 적용하는 이유는 모델이 특정 시점의 데이터뿐만 아니라 근접한 이전 구간의 모니터링 데이터를 종합적으로 판단하여 결과를 낼 수 있도록 하기 위함이다. 학습이 완료되면, 검증 데이터를 이용해 관련 연구[4]와 유사한 방법으로 임계값 기반의 접근 방식과 비교하여 학습된 모델의 성능을 평가한다.

최종적으로 도출되는 비정상 상태 탐지 시스템은 실시간으로 모니터링 데이터를 수집하고 학습 및 검증 시와 동일한 방법으로 전처리 과정을 거친 후, 적용된 모델에서 데이터가 비정상으로 분류되는 시점의 입력값과 비정상 여부의 분류 결과를 F1-Score 에 기반한 정확도와 함께 관리자 화면에 경보로 보여준다. 이를 통해 M-CORD 플랫폼 상에서 동작하는 LTE EPC 네트워크 구성요소의 비정상적인 상태를 인공지능을 이용하여 실시간으로 탐지한다.

Ⅳ. 결론

본 논문에서는 M-CORD 모니터링 시스템에서 수집되는 자원 사용 데이터를 머신러닝을 이용해 학습시켜 비정상 탐지 모델을 만들고, 학습된 모델을 시스템에 적용하여 M-CORD EPC 네트워크에 대한 비정상적인 상황을 탐지하는 시스템을 제안한다. 향후 연구로 본 비정상 상태 탐지 시스템을 구현하고, 머신러닝 기반의 비정상 상태 탐지의 실효성을 검증하며, 나아가 비정상 징후를 사전에 감지하는 예측 모델을 구현하여 발전시킬 예정이다.

ACKNOWLEDGMENT

이 논문은 2019 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2015-0-00575, 글로벌 SDN/NFV 공개소프트웨어 핵심 모듈/기능 개발).

참 고 문 헌

[1] L. Peterson et al., "Central office re-architected as a data center," IEEE Communications Magazine, vol. 54, no. 10, pp. 96-101, Oct. 2016.

[2] B. Han, V. Gopalakrishnan, L. Ji and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," IEEE Communications Magazine, vol. 53, no. 2, pp. 90-97, Feb. 2015.

[3] J. Hochenbaum, O. S. Vallis and A. Kejariwal, “Automatic Anomaly Detection in the Cloud via Statistical Learning,” arXiv:1704.07706v1, 2017.

[4] Chengwei Wang, V. Talwar, K. Schwan and P. Ranganathan, “Online detection of utility cloud anomalies using metric distributions,” in 2010 IEEE Network Operations and Management Symposium (NOMS), pp. 96-103, 2010.

[5] C. sauvanaud, K. Lazri, M. Kaâniche and K. Kanoun, “Anomaly Detection and Root Cause Localization in Virtual Network Functions,” in 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), pp. 196-206, 2016.

네트워크 텔레메트리를 활용한 머신 러닝 기반 네트워크 이상 탐지 기법 연구

남석현°*, 현종환*, 유재형† , 홍원기*

*포항공과대학교 컴퓨터공학과
† 포항공과대학교 정보통신대학원

{obiwan96, noraki, styoo, jwkhong}@postech.ac.kr

Machine Learning Based Anomaly Detection Using Network Telemetry

Sukhyun Nam°*, Jonghwan Hyun*, Jae-Hyoung Yoo† , James Won-Ki Hong*

*Department of Computer Science and Engineering, POSTECH
† Graduate School of Information Technology, POSTECH

요 약

네트워크 이상 탐지 기술은 네트워크 상의 플로우에 대한 정보를 수집하여 네트워크에서 발생하는 악의적인 공격을 실시간으로 탐지하는 기술이다. 실시간으로 패킷 단위의 세부적인 네트워크 정보를 제공하는 INT (In-band Network Telemetry) 를 이용하면 네트워크 이상 탐지 기술에서 실시간으로 더 세부적인 정보를 제공받을 수 있다. 본 논문에서는 INT 를 이용하여 추출한 네트워크 상태 정보를 머신 러닝의 입력 특징(feature)으로 이용하여 더 높은 정확도를 가진 네트워크 이상 탐지 시스템을 제안한다.

Ⅰ. 서 론

SDN (Software Defined Networking) 기술이 네트워크 관리의 새로운 패러다임으로 산업계의 관심을 받고 있다. SDN 은 데이터 평면(data plane)과 제어 평면(control plane)을 분리함으로서 네트워크 운용자가 상황에 맞게 통신 기능을 제어할 수 있도록 한다. 네트워크 공격을 쉽게 탐지하고 반응할 수 있는 SDN 의 이점에도 불구하고, 데이터 평면과 제어 평면의 분리로 인해 SDN 이 새로운 형태의 네트워크 공격의 대상이 되기도 한다 [1].

네트워크 상의 유해한 공격을 탐지하고 대응 하는 시스템인 NIDS (Network Intrusion Detection System) 는 크게 두 가지 방식을 따른다. 첫째는 시그니처 기반(signature-based)방법으로, 입력 데이터를 공격 데이터 셋의 시그니처(signagure)와 비교하여 판별하는 방법이다. 이는 수많은 NIDS 에서 사용되어 왔지만, 새로운 형태의 공격은 탐지하지 못하는 단점이 있다 [2]. 두 번째 방법은 이상 상태 기반(anomaly-based) 방법으로, 새로운 데이터를 정상 상태의 트래픽 데이터 셋과 비교하여 얼마나 정상에서 벗어나 있는지를 기준으로 판별하는 방법이다. 이상 상태 기반 방법은 데이터 셋에 없는 새로운 형태의 공격도 탐지해낼 수 있는 이점 때문에 최근에 연구가 활발히 진행되고 있으며, 특히나 최근 다양한 분야에서 괄목할 만한 성과를 내고 있는 머신 러닝 기술을 이용하면 더 우수한 성능을 낼 것으로 기대된다 [3].

이상 상태 기반 탐지 기법은 주로 플로우 기반(flow-based)으로 연구가 이루어지는데, 이는 패킷 헤더에 있는 정보만 이용하여 빠른 속도로 데이터를 처리하게 한다. 분류(classification) 기술은 데이터에서 추출하는 특징(feature)의 종류에 따라 성능이 크게 좌우되기 때문에 패킷 헤더에서 어떤 특징을 사용해야 탐지 정확도를 높일 수 있는지에 대한 연구가 많이 진행되었으며, 인입 포트 및 TCP 플래그 정보를 이용하면 정확도가 높음을 보였다 [2][4]. 최근에는 이에 더해 인입 포트의 엔트로피를 계산하여 높은 성능을 낸 연구가 진행되었다 [5].

본 연구에서는 머신 러닝을 이용하여 이상 상태 탐지 시스템을 설계하며, 입력 특징으로 플로우에서 추출하는 일반적인 특징에 더해 네트워크 텔레메트리에서 수집한 데이터를 사용하는 것을 제안한다. 네트워크 텔레메트리 기법 중 하나인 INT (In-band Network Telemetry)[6]는 실시간으로 네트워크의 정보를 패킷 단위로 제공하여 네트워크 이상 상태 감지의 입력 특징 생성에 적합하다.

본 논문에서는 프로그래머블 스위치로 구성된 SDN 환경에서의 INT 를 이용한 네트워크 이상 탐지 시스템 구조를 제안한다. 제안하는 네트워크 이상 탐지 시스템은 플로우 정보와 INT 를 이용해 추출한 네트워크 상태정보(메타데이터)를 머신 러닝의 입력으로 사용하여 더 높은 탐지 정확도를 가질 수 있다.

본 논문의 구성은 다음과 같다. 2 장에서는 기존의 네트워크 이상 탐지 기술과 INT 기술에 대해 기술한다. 3 장에서는 INT 를 활용한 네트워크 이상 탐지 구조를 제안한다. 마지막으로 4 장에서는 결론 및 향후 연구를 기술한다.

Ⅱ. 관련 연구

1. 네트워크 이상 탐지 기술

플로우 기반 이상 탐지 연구는 최근 폭 넓게 연구되고 있다. 새로운 데이터를 기존의 정상 상태 데이터 셋과 비교해야하기 때문에 classification 기법이 많이 쓰이는데, SVM (Support Vector Machine)을 이용하여 정확도 98.29%, MR (Miss Rate) 4.71%, FAR (False Alarm Rate) 0%를 기록한 연구가 있다 [4]. 해당 연구는 데이터 셋을 직접 제작하여 사용하였는데, 같은 데이터 셋에 대해 인공 신경망을 적용한 연구는 MR 0.6%, FAR 0.8%을 기록하였다 [2].

사생활 침해 등의 문제로 이상 상태 탐지 연구용 공개 데이터 셋은 매우 한정되어 있다. 그 중 NSL-KDD 데이터 셋이 가장 널리 쓰이는데, 이는 네트워크 이상 탐지 연구에서 널리 쓰이던 DARPA/KDD 데이터 셋이 너무 오래되어 최근의 네트워크와 특성이 달라 2014 년 KDD Cup 에서 새로 발표된 데이터 셋이다 [7]. 해당 데이터 셋의 플로우는 41 개의 특징을 가지고 있으며, 공격 유형은 크게 DoS (Denial of Service), R2L (Remote-to-Local), U2R (User-to-Root), Probe 공격으로 분류되는 22 개의 공격을 담고 있다. 특히, 일부 공격은 테스트 데이터 셋에만 포함되어 있고 트레이닝 데이터 셋에는 포함되어있지 않아 실제 네트워크를 잘 표방하고 있다.

NSL-KDD 데이터 셋에서 포함하고 있는 네 유형의 공격 중 DoS 공격을 제외한 Probe, R2L, U2R 은 트래픽의 특성이 일반 트래픽과 크게 다르지 않아 이상 상태 기반으로는 감지하기 힘들다 [8]. 이 때문에 DoS 만 대상으로 하여 높은 성능을 기록한 연구가 존재한다 [9]. 딥 러닝을 접목하여 모든 공격에 대해 높은 정확도를 갖기 위한 연구가 있으나, F1 score 75.75%로 앞선 연구들에 비해 낮은 성능을 기록하였다 [10].

2. In-band 네트워크 텔레메트리 (INT) [6]

네트워크 텔레메트리는 현재의 네트워크는 장비의 개수, 트래픽 양이 과거와 달리 매우 복잡해 그동안의 방법으로 네트워크 트래픽 정보를 얻기는 힘들기 때문에 제안된 방법으로, 네트워크에서 생성되는 다양한 정보들을 수집하는 것을 의미한다. 그 중 In-band 방식은 별도의 탐지 패킷을 생성하지 않고 기존 데이터 트래픽에 텔레메트리 데이터를 삽입하는 방식이다.

INT 는 P4 를 활용한 네트워크 모니터링 프레임워크로, In-band 방식을 이용하기 때문에 프로그래머블 스위치로 이루어진 네트워크 상에서 제어 평면의 개입 없이 데이터 평면에서 네트워크의 상태를 수집할 수 있다.

네트워크 상에서 INT 정보가 추출되는 방식은 다음과 같다. INT 를 지원하는 스위치는 그 역할에 따라 Source 스위치, Transit 스위치, Sink 스위치로 구분된다. Source 스위치는 수집하고자 하는 네트워크 정보의 종류를 패킷의 INT 헤더에 삽입하여 전달한다. Transit 스위치는 패킷에 INT 헤더가 포함되어 있는지 확인하여 INT 헤더가 포함되어있을 경우, 이를 읽어 들여 상응하는 네트워크 상태정보를 패킷에 삽입하여 전달한다. Sink 스위치는 INT 헤더 및 상태 정보를 추출한다.

INT 를 통해 수집 가능한 네트워크 상태 정보는 스위치 ID, 인입 포트 관련 정보, 인출 포트 관련 정보, 버퍼 관련 정보 등이 정의되어 있다. INT 를 사용하면 기존의 네트워크 모니터링 기법과는 달리 실시간으로 패킷 단위의 세부적인 네트워크 정보를 수집할 수 있어 네트워크 가시성을 최대한으로 제공해 준다.

Ⅲ. 이상 상태 탐지 시스템 구조

본 장에서는 프로그래머블 스위치로 구성된 SDN 네트워크 환경에서 INT 데이터를 이용해 이상 상태를 탐지하고 이를 반영하여 네트워크를 관리하기 위한 구조 및 방법에 대해 기술한다. SDN 네트워크에 기존의 제어 평면과 데이터 평면 외에 INT 데이터 관리를 위한 관리 평면과 이상 탐지를 실행하는 지식 평면[11]이 추가되었다 (그림 1).

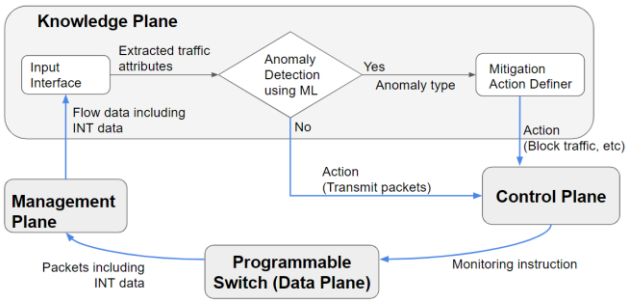


그림 1 이상 탐지 기법을 적용시킨 SDN 네트워크 구조

각 평면 별 기능은 다음과 같다.

1. 제어 평면 (Control Plane)

SDN 컨트롤러가 제어 평면의 역할을 수행한다. 프로그래머블 스위치로 구성된 데이터 평면에 INT 프로그램을 배포함과 동시에 지식 평면에서 생성된 네트워크 요구사항 반영 등의 기능을 수행할 수 있다.

2. 데이터 평면 (Data Plane)

데이터 평면에서는 INT 메타데이터 생성, 추출 및 전송의 기능을 수행한다. 제어 평면에서 배포된 INT 프로그램에 따라 각 스위치에서 해당하는 INT 메타데이터를 패킷 헤더에 추가하여 전달한다.

3. 관리 평면 (Management Plane)

관리 평면에서는 스위치로부터 전송된 INT 메타데이터를 수집하여 지식 평면으로 전달하기 위해 분석하는 기능을 수행한다. 패킷 헤더에서 패킷 정보 및 INT 메타데이터를 추출한 후 플로우 별 통계 정보를 작성하여 지식 평면으로 전달한다.

4. 지식 평면 (Knowledge Plane)

지식 평면에서는 관리 평면에서 전달된 플로우 데이터를 이용하여 이상 상태 탐지를 수행한다. 지식 평면은 데이터 셋을 이용하여 사전에 학습한 머신 러닝 알고리즘을 포함하고 있다. 해당 머신 러닝 알고리즘은 데이터 셋으로부터 이상 상태 탐지에 적합한 특징을 추출하여 학습시킨 간단한 심층 신경망(DNN, Deep Neural Network) 이다. 지식 평면에서는 플로우 데이터를 머신 러닝의 입력 행렬로 변환한다. DNN 이 각 플로우 별 이상 상태 점수를 생성해낸다. 지식 평면은 임계치를 기준으로 이상 상태인지 아닌지를 판별한다. 만약 이상 상태가 아니라면 제어 평면에 해당 플로우는 정상이니 INT 메타 정보를 추출할 필요 없다는 정보를 전달 한다. 이상 상태가 감지되면 지식 평면 내의 완화 작용 선택 과정을 거쳐 적절한 완화 작용(플로우 차단, 로드 밸런싱, 발신 포트 차단 등)을 수행하도록 전달한다.

Ⅳ. 결론 및 향후 연구

INT 는 기존의 네트워크 모니터링 기법과는 달리 실시간으로 패킷 단위의 세부적인 네트워크 정보를 수집할 수 있어 네트워크에 대한 더 많은 정보를 실시간으로 얻을 수 있다. 본 연구에서는 해당 네트워크 텔레메트리를 활용해 네트워크 이상 상태 탐지에서 높은 성능을 낼 수 있는 방안을 논의하였다. INT 를 머신 러닝에 적용시킨다면 기존의 연구들보다 더 높은 성능을 낼 수 있을 것으로 기대된다. 더해서 기존의 DoS 를 주 대상으로 탐지해낸 연구들과 달리 R2L 과 L2U 같이 정상 상태 패킷과 별다른 차이가 없어 이상 상태 기반 방법으로는 탐지할 수 없다고 알려진 공격들도 탐지해낼 것으로 기대된다.

ACKNOWLEDGMENT

이 논문은 2019 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (2018-0-00749, 인공지능 기반 가상 네트워크 관리기술 개발)

참 고 문 헌

[1] D. Kreutz, F. Ramos, and P. Verissimo, “Towards secure and dependable software-defined networks,” Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. ACM, 2013, pp. 55- 60.

[2] Z. Jadidi, V. Muthukkumarasamy, E. Sithirasenan, and M. Sheikhan, “Flow-Based Anomaly Detection Using Neural Network Optimized with GSA Algorithm,” IEEE 33rd International Conference on Distributed Computing Systems Workshops, 2013, pp. 76-81.

[3] R. Sommer, and V. Paxson, “Outside the Closed World: On Using Machine Learning for Network Intrusion Detection,” IEEE Symposium on Security and Privacy, 2010, pp. 305-316.

[4] P. Winter, E. Hermann, and Z. Markus, “Inductive Intrusion Detection in Flow-Based Network Data using One-Class Support Vector Machines”, IFIP International Conference on New Technologies, 2011, pp. 1-5.

[5] L. F. Carvalho, T. Abrão, L. S. Mendes, and M. L. Proença, “An ecosystem for anomaly detection and mitigation in software-defined networking,” Expert Systems with Applications, Volume 104, 2018, pp. 121-133.

[6] C. Kim, A. Sivaraman, N. Katta, A. Bas, A. Dixit, and L. J. Wobker, “In-band Network Telemetry via Programmable Dataplanes,” ACM SOSR, 2015, pp. 2-3.

[7] “NSL-KDD,” 2014, (https://www.unb.ca/cic/datasets/nsl.html).

[8] M. Ahmed, A. Mahmood, and J. Hu, “A survey of network anomaly detection techniques,” Journal of Network and Computer Applications, Volume 60, 2016, pp. 19-31.

[9] R. Braga, E. Mota, and A. Passito, “Lightweight DDoS flooding attack detection using NOX/OpenFlow,” IEEE Local Computer Network Conference, 2010, pp. 408-415.

[10] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, “Deep learning approach for Network Intrusion Detection in Software Defined Networking,” WINCOM, 2016, pp. 258-263.

[11] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, “A knowledge plane for the internet,” SIGCOMM, 2003, pp. 3-10.

인공지능 기반 VNF 자원 예측 모델 연구

김희곤*○, 정세연*, 유재형† , 홍원기*

*포항공과대학교 컴퓨터공학과

† 포항공과대학교 정보통신대학원

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

A Study on Machine Learning based VNF Resource Prediction

Hee-Gon Kim*○, Se-Yeon Jeong*, Jae-Hyung Yoo† ,James Won-Ki Hong*

*Department of Computer Science and Engineering, POSTECH

† Graduate school of Information Technology, POSTECH

요 약

네트워크 기능 가상화 (Network Function Virtualization, NFV)는 네트워크를 구성하는 주요 요소들을 소프트웨어로 구현하여 가상화된 환경에서 운용하고자 하는 것으로서, 환경 변화에 따라 동적으로 가상 네트워크 기능 (VNF)을 관리함으로써 네트워크 투자비용과 운용비용을 절감할 수 있다. 본 논문에서는 자동화된 VNF 관리를 위해, 기계 학습을 사용한 VNF 자원 사용 예측 모델을 제안하고 이를 검증하기 위해 동적인 Service Function Chain (SFC)에 적용한 결과를 보인다. 제안한 모델은 기존의 기계학습 모델과 비교해 11% 높은 예측 성능을 보이며, 서로 다른 토폴로지의 네트워크에도 적용 가능하다는 장점이 있다.

Ⅰ. 서 론

네트워크 기능 가상화(Network Function Virtualization, NFV) 의 출현은 네트워크의 새로운 패러다임을 만들었다. 가상화 환경에서 네트워크 기능을 구현하며 제어하는 것으로 네트워크 관리자는 환경 변화에 따라 동적으로 네트워크 관리를 할 수 있게 되었으며 보다 효율적인 네트워크 운용이 가능해졌다. NFV 환경에서는 기존의 물리 네트워크를 여러 개의 가상 네트워크 기능(Virtual Network Function, VNF)으로 구현하고 이들을 조합하여 서비스를 제공할 수 있다.

관리자는 동적으로 VNF 의 자원 사용량을 제어 할 수 있으며, 이는 네트워크의 상태에 따라 적절하게 자원을 Scale in/out 하여 보다 효율적인 관리를 할 수 있음을 말한다. 하지만 이러한 VNF 의 장점에도 불구하고 실제의 네트워크는 복잡하며 빠르게 변화하는 특성을 가지고 있어 단순한 관리 방법으로는 최적화된 네트워크 관리가 어려우며 전문적인 지식을 가진 인력을 필요로 한다.

기계 학습(Machine Learning)은 동적으로 변화하는 네트워크를 관리하는 차세대 방법으로 주목을 받고 있다 [1]. 기계 학습은 기계가 주어진 데이터를 학습하여 최적화된 행동 혹은 데이터를 도출하는 방법이다. 기계 학습을 VNF 자원 예측에 사용하는 것으로, 관리자는 NFV 네트워크의 상태를 예측하며 최적화된 관리를 수행할 수 있게 된다.

본 논문은 기계 학습을 사용하여 VNF 자원 예측을 수행한다. 제시하는 모델은 LSTM [2], Attention [3] 등 다양한 기계학습 기법을 사용하며, Service Function Chaining 정보를 사용하는 것으로 개별 VNF 의 자원 예측 성능을 높인다 [4,5]. 또한 본 논문에서 제시하는 모델은 범용성을 가지도록 설계하여 여러 네트워크 토폴로지에서도 학습이 가능하도록 하였다.

Ⅱ. 본 론

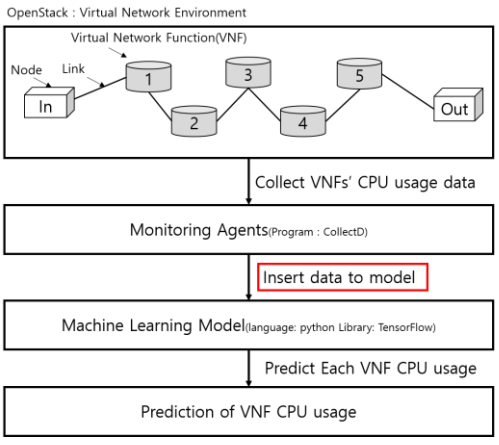


그림 1 VNF 자원 예측 시스템 도안

이번 연구에서는 그림 1 의 도식 대로 VNF 자원 사용을 예측하였다. Openstack [6] 환경에서 5 개의 Node 와 2 개의 In-Out Node 로 구성된 네트워크에서 각 VNF 들의 자원을 Collectd [7] 를 이용하여 12 시간 동안 5 초간격으로 수집하였다. VNF 1-5 는 순서대로

Firewall, NAT, IPS, DPI, Load Balancer 기능을 수행한다. vCPU 1 개와 vRAM 2 개가 VNF 들의 기본 Deployment 설정이며, IPS 와 DPI 의 경우 각 자원을 2 배로 할당하였다. 우리는 webserver stress tool 을 사용하여 In-node 에서 out-node 로 traffic 을 생성하여 보냈다.

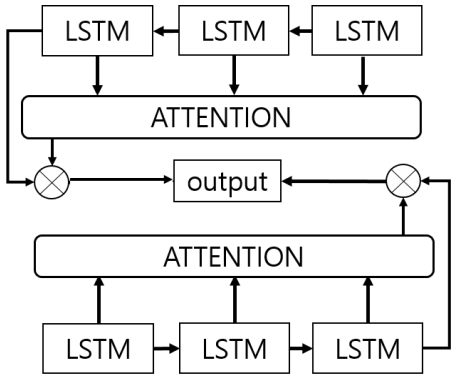


그림 2 VNF 자원 예측 모델

본 연구에서는 Long Short Term Memory (LSTM) 과 Attention 을 활용하여 VNF 자원 사용을 예측하였으며 학습 모델로 기본 LSTM 과 더불어 CAT-LSTM, AT-Bi-directional LSTM 을 사용하였다. 그림 2 는 CAT-LSTM 과 AT-Bi-directional LSTM 모델을 간단히 도식한 것이다. 학습 모델은 각 모델의 기준에 따라 입력 데이터를 분리하여 사용하였으며, 이로 인해 높은 범용성을 가지게 되었다. 본 논문의 모델은 동시에 여러 VNF 들의 학습이 가능하며 서로 다른 토폴로지를 가지는 네트워크의 학습도 가능해졌다.

Ⅲ. 실험

본 논문에서는 각 VNF 의 CPU 사용량을 Root Mean Square Error (RMSE) 손실 함수를 사용하여 예측하였다. 0.01 의 학습률 (Learning rate) 을 사용하여 500 번의 Iteration 결과, 기본 LSTM 과 CAT-LSTM, AT-Bi-directional LSTM 은 각각 7.0, 6.2, 6.1 의 loss 를 가지게 되었다. 이 결과는 CAT-LSTM 과 AT-Bi-directional LSTM 이 기본 LSTM 보다 성능이 뛰어남을 말하며 Attention 사용의 중요성을 말한다.

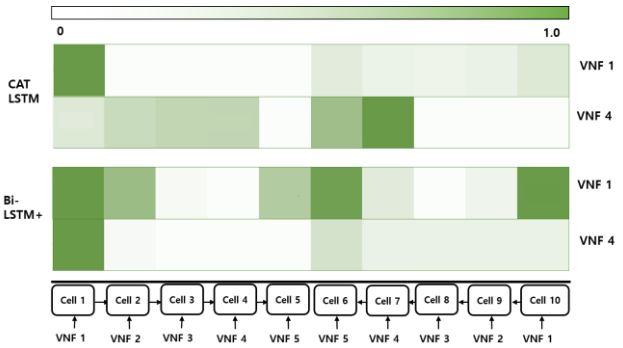


그림 3 Attention 학습 결과

그림 3 은 CAT-LSTM 과 AT-Bi-directional LSTM 의 Attention 학습 정보를 나타낸다. Attention 은 입력 데이터의 어떤 특정 데이터가 결과 데이터에 높은 연관성을 가지게 되는 지 학습을 하며, 이때 얻은 정보를 토대로 학습 데이터에 가중치를 둔다. 그림 3 은 Attention 이 학습한 가중치를 표현한 것이다. 그림 3 에서 짙은 색을 가질수록 높은 가중치를 가지며 하얀색을 가질 수록 낮은 가중치를 가지게 된다. 각 모델은 그림 3 에서 제시된 것과 같이 VNF 1-5 의 데이터를 입력데이터로 받은 뒤에 VNF 1 과 VNF 4 의 자원 사용을 예측한다. 이때 CAT-LSTM 은 SFC 를 이루는 주위의 VNF 데이터로부터 학습을 하기보다는 자기 자신의 데이터에 가중치를 많이 두어 학습하는 경향이 있으며, AT-Bi-directional LSTM 의 경우는 상대적으로 주위의 VNF 데이터에 가중치를 두는 모습을 볼 수 있다. 이는 모델이 가진 구조의 차이를 보여주는 것이며, 특정 VNF 가 주위의 VNF 에 영향을 많이 받는지 특성을 고려하여 선택적으로 학습 모델을 사용한다면 더 높은 학습결과를 만들 수 있다는 것을 의미한다.

Ⅴ. 결론

본 논문에서는 기계 학습을 사용하여 VNF 자원을 예측하였다. 제안하는 모델은 SFC 를 입력데이터로 사용하였으며, 다양한 머신 러닝 기법을 사용하였다. 본 논문에서 제안한 학습 모델은 기본 LSTM 모델을 사용하였을 때 보다 11% 높은 정확도를 가졌다. 이는 본 논문에서 제안하는 방법이 동적으로 네트워크를 관리하는데 도움이 되며 운용비용을 줄일 수 있음을 가리킨다. 우리는 Simulator 가 아닌 OpenStack 기반의 테스트베드를 사용하여 다른 모델과의 예측 성능을 비교하였다.

ACKNOWLEDGMENT

본 연구는 2019 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (2018-0-00749, 인공지능 기반 가상 네트워크 관리기술 개발, IITP-2019-2017-0-01633* 대학 ICT 연구센터지원사업)

참 고 문 헌

[1] R. Boutaba, M. A. Salahuddin, N. Limam, et al. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. Journal of Internet Services and Applications, 2018

[2] S. Hochreiter and Jurgen Schmidhuber. Long short-term memory. Neural Comput., 1997.

[3] A. Vaswani, N. Shazeer, N. Parmar, et al. Attention is all you need. CoRR, 2017

[4] 기계학습 기반의 가상 네트워크 기능 자원 수요 예측 방법 (A Machine Learning-based Method for Virtual Network Function Resource Demand Prediction) 김희곤, 이도영, 유재형, 홍원기, KNOM Review, Vol. 21, No. 2, Dec. 2018, pp. 1-9.

[5] D. Tang, B. Qin, X. Feng, et al. Effective LSTMs for Target-Dependent Sentiment Classification. In International Conference on Computational Linguistics (COLING), 2016.

[6] N. McKeown, T. Anderson, H. Balakrishnan, et al. Openflow: Enabling innovation in campus networks. Computer Communication Review, 2008

[7] F. Forster et al. collectd the system statistics collection daemon, 2012

가청 주파수를 이용한 Hello 보내기

김수현*, 문현수, 이영석

충남대학교

{shkim95,munhyunsu,lee}@cnu.ac.kr

Data Transmission Experiment Using Audio Frequency

Soohyun Kim*, Hyunsu Mun, Youngseok Lee

요 약

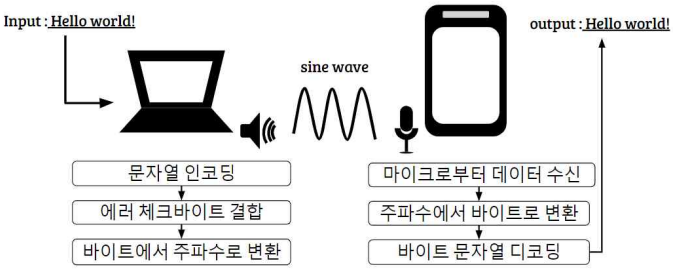
데이터 오버 사운드(Data Over Sound)는 소리로 데이터를 보내는 통신기술이다. 데이터를 보내는 기술은 네트워크 어댑터가 없어도 기존에 있는 마이크와 스피커로 소리를 보낼 수 있다. 스타벅스의 사이렌 오더는 소리로 데이터를 보내는 대표적인 상용 서비스이며 이외에도 Chirp.io, LINSR등 소리로 데이터를 전송하는 프로그램이 많이 등장하고 있다. 본 논문은 디바이스에서 제약이 없는 소리 송수신 프로그램을 제작하였고, 실험 결과 40cm 이내의 거리까지는 80%의 정확도를 보였으며 그 이상의 거리로 데이터를 이용한 문자열을 송신할시 급격하게 정확도가 낮아지는 것을 확인할 수 있었다.

Ⅰ. 서 론

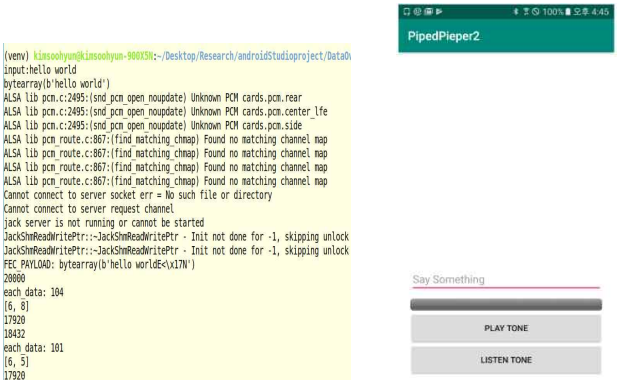
데이터 오버 사운드(Data Over Sound)는 소리로 데이터를 보내는 통신 기술로, 와이파이나 블루투스로 어댑터가 추가로 필요하지 않아도 기존의 마이크와 스피커만 있으면 근거리 통신이 가능하다[1]. [2]는 스피커와 마이크만 이용하여 NFC를 만들었다. 상용 서비스 중 스타벅스의 사이렌 오더[1] 기술은 소리로 통신하는 대표적인 기술이다. 사이렌 오더는 스타벅스 애플리케이션을 실행하여 음료를 주문할 때 고객의 실내위치를 파악하는데 사용된다. LISNR[2] 역시 소리로 데이터를 송수신하는 프로그램으로, 상용화를 준비 중인 서비스이다. LISNR는 소리 통신을 다양한 분야에서 활용하는데, 데이터 송수신뿐만 아니라, 소리로 문을 여는 프로그램이나, 티켓팅에서 활용될 수 있음을 보인다. Chirp[3]은 스마트폰 애플리케이션에서 사용되는 소리 채팅 프로그램으로, 문자를 입력하고 전송 버튼을 누르면, 가청 주파수 대역에서 휴대폰간의 데이터를 송수신 가능하다. 소리로 결제하는 서비스 모비두[4]는 롯데의 결제서비스 L.PAY에 활용되어 어댑터의 호환 문제없이 어디서든 결제가 가능하다.

본 논문에서는 디바이스에 제약이 없는 소리 송수신 프로그램을 제작하였다. 데이터 송수신시 비트 오류가 발생할 수 있다. 따라서, 데이터가 중간에 변경되거나 손실되는 오류를 검출해야한다. 이를 위하여 오류 검출 알고리즘을 포함한 소리 송수신 프로그램을 제안한다. [5]

Ⅱ. 소리 통신 프로그램



(그림 1) 소리로 데이터를 주고받는 프로그램의 구성도



(그림 2) 가청주파수를 이용한 송수신 프로그램 Linux Mint 노트북 송신(좌) 안드로이드 어플리케이션 수신 (우)

Handshake Start Hz	Handshake End Hz	StepHz	StartHz
4096	6144	256	1024

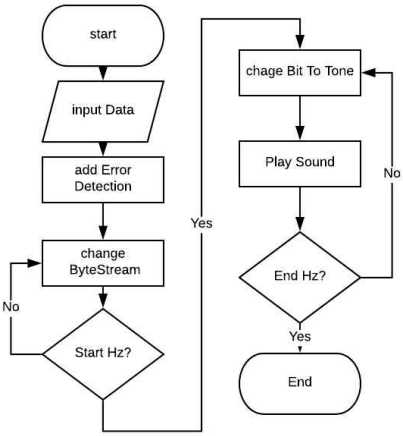
(표 1) 소리 통신을 위한 주파수(Hz)

소리로 데이터를 주고받기 위해서 마이크와 스피커를 사용하였으며, 가청주파수 대역 20~20000Hz 사이의 주파수를 이용 하였다. 수신기가 수신 주파수를 인식하기 위하여 송수신시 송신 시작 주파수와 송신 끝 주파수 및 각 톤의 구분을 위한 주파수를 표1과 같이 미리 정의하였다. Handshake Start Hz는 송수신 시작을 위한 시작 주파수이다. Handshake End Hz는 송수신 종료를 위한 끝 주파수이다. StepHz는 하나의 톤을 구분해주기위한 주파수이다. StartHz 한 톤의 시작 주파수를 의미한다.

2.1송신기

송신프로그램은 리눅스 기반 파이썬 프로그램이다. 송신프로그램이 사용자로부터 데이터를 입력받으면, 입력받은 데이터에 비트 오류 검출하는 알고리즘인 리드-솔로몬 코드를 이용하여 에러검출용 4바이트를 추가하여 코드 워드를 만든다. 만들어진 코드 워드를 주파수로 변환하기 위하여 각 코드 워드에서 한 바이트를 4비트로 나누어 총 16개의 톤을 표현하였다. 16개의 톤을 이용하여 주파수를 생성하는 공식은 표 1과 같이 시작 주

1)https://www.mk.co.kr/news/business/view/2017/01/62241/
2) https://lisnr.com/
3) https://chirp.io/
4) https://www.venturesquare.net/770906
5) https://youtu.be/FdD1SYUS55E



(그림 3) 송신기측 전송 순서도

파수부터 각 톤을 구분해주기위한 스텝 주파수와 톤을 곱해주어 최종 주파수를 만들었다.

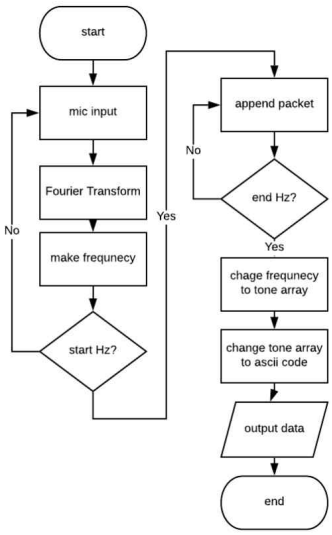
$$\text{Frequency} = \text{StartHz} + \text{StepHz} * \text{Tone} \cdots\cdots\cdots\cdots (1)$$

모든 데이터가 주파수로 매핑이 되어 배열로 저장을 하면, 주파수에 따른 신호를 생성해주었다.

$$f(x) = A\sin(2\pi f x + \theta) \cdots\cdots\cdots\cdots (2)$$

수식 2처럼, 사인 함수를 이용하여 주파수를 소리 신호로 바꾸었다. 여기서 A는 f는 주파수이고, θ는 음파에 대한 위상이다. 소리를 실제로 생성하려면 수식(2)을 그대로 사용하면 연속된 함수이기 때문에 실제 소리가 생성되지 않는다. 따라서 소리를 출력하기 위해서 사인파에서 샘플링을 해서 이산적인 값으로 표현해주어야 한다. 따라서 44100개의 샘플 개수로 나누었다.

2.2수신기



(그림 4) 수신기 측 수신 순서도

수신 프로그램은 안드로이드 휴대폰 기반의 애플리케이션으로 제작하였다. 송신 측 프로그램으로부터 송신한 음파는 안드로이드 휴대폰의 마이크로부터 데이터를 수신받는다. 초기 어플리케이션으로부터 수신 받은 값은 송신측 프로그램으로부터 사인함수를 샘플링한 정보로, 샘플링한 값을 주파수로 바꾸어 주어야한다. 수신받은 샘플링 값은 주파수정보로 변환하기 위하여 주기함수들로 바꿔주어야한다. 수식 (3)처럼 주기함수로 바꾸어주는 과정을 이산 푸리에 변환을 이용하였다. 수식 (3)에서 N은으로 0부터 N-1로 표현이된다. k는 주파수 함수 성분의 인덱스이다.

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-2\pi kn/N}, k = [0, N-1] \cdots\cdots\cdots\cdots (3)$$

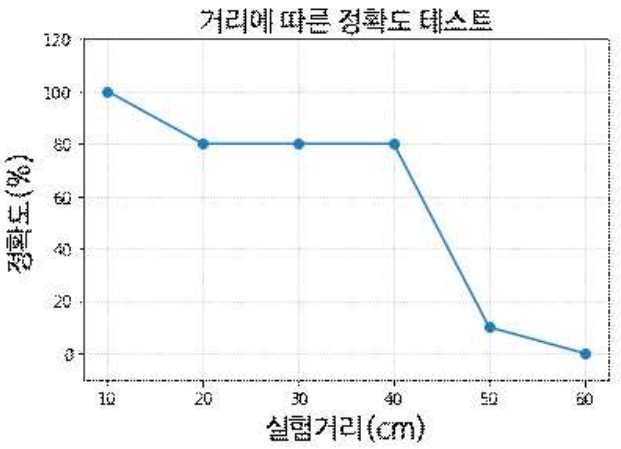
이산 푸리에 변환을 통해 주파수로 바꿔주면, 변환된 주파수를 이용하여 통신 시작 주파수와 통신 끝 주파수 구간의 주파수들을 이용하여 원래 문자열로 변환해준다. 송신측에서 문자를 주파수로 변환한 형태의 반대과정을 통하여 실제 문자열로 바꿔준다.

$$\text{Tone} = \text{Frequency} / \text{StepHz} - \text{StartHz} \cdots\cdots\cdots\cdots (4)$$

수식(4)를 통하여 아스키 코드를 4비트로 분할한 비트를 한 바이트로 합쳐 아스키 코드로 만들어 준다. 만들어진 아스키 코드들로부터, 리드 - 솔로몬 에러 탐지 비트를 계산하여 송신중 잘못된 문자들을 검출 한다.

Ⅲ. 실 험

만들어진 프그램의 테스트를 위하여 Galaxy A5 모델을, 수신기로는 송신기로는 Linux Mint를 시용하였다. 송수신기간의 거리를 이용하여 각 10번씩 실험하였다. 표1 에서 제시된 것처럼 송신기와 수신기 사이의 거리를 10cm씩 늘여가면서 실험을 진행하였다. 가장 가까웠던 10cm는 10번 중 모든 데이터를 오류 없이 정확하게 받을 수 있었으며, 특정 거리 이상이면 정확도가 급격히 떨어져 10개 중 1개의 데이터만 받을 수 있었으며, 50cm이상의 거리에서 실험했을 때는 어떤 데이터도 받을 수 없었다. 실험거리가 멀어질수록 오류를 검출하는 횟수가 더 많아짐을 확인할 수 있었다.



(그림 5) 거리에 따른 정확도 테스트

Ⅳ. 결론

본 논문은 안드로이드 휴대폰과, 리눅스 기반 컴퓨터로 소리로 데이터를 송수신하는 프로그램을 보였다. 현재 오류검출 알고리즘인 리드-솔로몬은 최대 33비트의 데이터만 에러만 검출 할 수 있으며, 오류 수정은 불가능하다. 따라서 오류 수정 알고리즘 및 4바이트가 넘어가는 긴 파일을 전송할 때 가변길이 오류 검출 코드를 사용한다면, 간단한 문자뿐만 아니라 파일전송 등 대량의 데이터를 송수신 가능할 것으로 보인다.

ACKNOWLEDGMENT

이 논문은 2016년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (NRF-2016R1D1A1A09916326)

참 고 문 헌

[1]이형진, 문현수, 이영석,(2016).비콘의 속삭임 : 소리를 이용한 ASCII 코드 전송 애플리케이션.한국정보과학회 학술발표논문집,(),1480-1482.

[2] Rajalakshmi Nandakumar, Krishna Kant Chintalapudi, Venkat Padmanabhan, and Ramarathnam Venkatesan. 2013. Dhwani: secure peer-to-peer acoustic NFC. In Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM (SIGCOMM '13). ACM, New York, NY, USA, 63-74

머신러닝 기반 동적 경로 가중치 조정 로드밸런싱 알고리즘 연구

임지윤°*, 현종환*, 유재형† 홍원기*

*포항공과대학교 컴퓨터공학과
†포항공과대학교 정보통신대학원

{limjiyoon, noraki, styoo, jwkhong}@postech.ac.kr

A Study on Machine learning-based Dynamic path weight adjustment load balance algorithm

Jiyoon Lim°*, Jonghwan Hyun *, Jae-Hyoung Yoo†, James Won-Ki Hong*

*Department of Computer Science and Engineering, POSTECH
†Graduate School of Information Technology, POSTECH

요 약

데이터센터에서 로드밸런싱 알고리즘을 통해 네트워크의 성능을 잘 활용할 수 있다. 하지만 데이터센터의 트래픽은 기존 네트워크 트래픽과 다른 특성을 가지고 있기 때문에 데이터 센터를 위한 로드밸런싱 알고리즘들이 많이 제안되고 있다. 기존에 많이 사용하고 있는 ECMP 알고리즘의 경우 대용량 플로우가 흐를 경우 제대로 부하를 분산시키지 못한다는 한계점이 있다. 이를 해결하기 위해 중앙 집중화된 알고리즘이 연구되었지만 중앙 집중화된 컨트롤러의 오버헤드 문제가 발생한다. 본 연구는 중앙 집중화된 컨트롤러에 부담이 적고 네트워크 변동에 빠른 반응성을 갖는 머신러닝 기반 동적 경로 가중치 조정 로드밸런싱 알고리즘을 제안한다. 제안하는 알고리즘은 P4 와 In-band 네트워크 텔레메트리를 통해 높은 네트워크 가시성을 확보한다. 그리고 머신러닝을 통해 트래픽 패턴을 예측하여 경로들의 가중치를 계산해 기존 중앙 집중화된 알고리즘보다 빠른 패킷 처리를 기대할 수 있다.

Ⅰ. 서 론

인터넷 서비스들이 클라우드 환경에서 제공되는 경우가 많아지면서 데이터센터는 각 서비스들을 빠르고 안정적으로 제공하기 위해 높은 경로다양성을 갖는 토폴로지를 사용한다[1]. 이러한 토폴로지들은 높은 bisection 대역폭을 가지며, 이를 최대한 활용하여 네트워크의 이용률을 높이기 위해서 로드밸런싱 알고리즘이 사용된다[2]. 하지만 데이터 센터는 인터넷에 비해 토폴로지의 변동이 심하지 않고 트래픽이 빠르게 변동하는 특성을 갖는다[2][3]. 따라서 데이터센터에서는 기존의 로드밸런싱 알고리즘과는 다른 데이터 센터의 특성에 맞는 로드밸런싱 알고리즘을 사용해야 한다.

데이터센터에서 주로 사용하고 있는 로드밸런싱 알고리즘으로는 ECMP (Equal-cost Multipath) [4]가 있다. ECMP 는 출발지에서 목적지까지의 최단 경로가 복수개일 경우, 정적 해싱을 통해 트래픽을 각 최단경로에 균등하게 분배한다. 하지만 ECMP 의 경우 데이터센터에 적용하기에 한계를 갖고 있다. 그 중 하나는 둘 이상의 대용량 트래픽이 같은 경로로 할당되어 제대로 부하가 분산이 되지 않을 수 있다는 것이다[4][5][6].

ECMP 알고리즘의 한계를 해결하기 위해 네트워크의 상태 정보를 파악하고 이를 활용하는 로드밸런싱 알고리즘들이 제시되었다. Hedera[5]는 하나의 중앙 집중화된 컨트롤러가 주기적으로 각 스위치에서의 플로우의 양을 측정한다. 그리고 대용량 플로우가 측정될 경우 경로의 용량이 해당 대용량 플로우를 수용할 수 있는 경로 중에서 미니맥스 결정방식으로 할당한다. 이렇게 중앙화된 컨트롤러에서 라우팅을 하는 경우 부하를 분산시킬 수 있는 최적의 경로로 배정할 수 있다는 장점이 있다. 하지만 대량의 트래픽이 짧은 시간에 발생하는 경우 컨트롤러에서 패킷 처리가 지연되는 단점이 있다[6][7]. 데이터 센터에서 대부분의 트래픽 정체는 대량의 트래픽에 의해 발생하기 때문에 이는 성능에 심각한 영향을 주게 된다[2][3]. 따라서 이러한 대량의 트래픽이 발생했을 때에도 빠르게 패킷을 처리하여 적절한 경로로 패킷을 보낼 수 있도록 하기 위한 연구가 많이 진행되고 있다[7][8][9].

정밀한 네트워크 상태 정보를 수집하기 위해 P4[10]와 프로그래머블 스위치를 사용하기도 한다[7][8]. P4 는 프로그래머블 데이터 평면을 위한 DSL (Domain-specific language)이다. P4 프로그램은 프로토콜의 헤더 포맷 정의, 헤더 파싱, match-action 테이블 구조 및 컨트롤 플로우를 프로그래밍할 수 있다. 스위치로 들어오는 패킷의 헤더를 정의된 규칙에 따라

파싱하고, match-action 테이블을 거치면서 헤더를 변경하며 패킷을 재조립하여 출력 포트로 내보내는 것이 가능하다.

In-band 네트워크 텔레메트리(INT)[11]는 P4 의 기능을 이용한 패킷 단위의 네트워크 모니터링 기법이다. 이 기법을 통해 수집 가능한 네트워크 상태 정보는 포트 관련 정보, 버퍼 관련정보 등이 있다. In-band 네트워크 텔레메트리는 제어 평면의 개입 없이 네트워크의 상태 정보를 수집할 수 있다는 장점이 있다. 또한 각 스위치마다 모든 패킷으로부터 네트워크 상태 정보를 수집할 수 있어 최대한의 네트워크 가시성을 제공한다는 장점이 있다.

최근에는 머신러닝을 활용해 트래픽을 예측하는 알고리즘들이 제시되었다[12]. 이러한 알고리즘을 활용하여 대용량 플로우를 미리 예측하여 처리할 경우 기존의 방법들보다 네트워크 변동에 빠르게 반응할 수 있을 것으로 기대된다.

본 논문은 P4 를 활용한 In-band 네트워크 텔레메트리 기법을 통해 네트워크 상태 정보를 수집하고 머신러닝 기법을 활용해 트래픽 패턴을 예측하여 이에 따라 동적으로 경로 가중치를 결정해 로드밸런싱 하는 방법을 제안한다. P4 와 In-band 네트워크 텔레메트리를 통해 기존의 방식보다 효율적으로 더 많은 네트워크 상태 정보를 수집한다. 그리고 수집된 정보를 머신러닝으로 학습시켜 예측한 트래픽 패턴에 맞게 동적으로 로드밸런싱 알고리즘을 적용시켜 네트워크 전체의 평균 처리율(average throughput)을 향상시키고자 한다.

Ⅱ. 관련 연구

중앙 집중화된 컨트롤러에서 모든 정보를 처리하고 경로를 선택하는 방법의 경우 컨트롤러에 많은 부하가 발생하는 단점이 있다. 다음은 이를 해결하기 위해 각 스위치나 호스트에서 네트워크 상태 정보를 수집하고 최적 경로를 선택해 네트워크의 트래픽 변화에 빠르게 대응하는 연구들이다.

CONGA[7]는 네트워크 상태 변화에 빠르게 대응하기 위해 로드밸런싱 알고리즘을 각 스위치에서 실행하는 분산화된 구조를 제안했다. 패킷에 트래픽 혼잡 정보를 저장할 수 있는 헤더를 추가시켜 네트워크 상태 정보를 파악한다. 모든 leaf 스위치는 다른 leaf 스위치까지의 트래픽 혼잡 정보를 저장한다. 이후 패킷을 보낼 경우 플로우를 flowlet[13] 단위로 나누어 가장 혼잡 정도가 적은 다음 경로를 지정한다. CONGA 는 중앙 집중된 방식에서 초 단위로 반응했던 것에 비해 마이크로 초 단위로 반응한다는 장점이 있다. 또한 flowlet 단위로 경로를 할당함으로써 패킷 재배열 문제를 해결했다[14]. 하지만 자체 제작한 스위치를 사용했고, 2-tier 토폴로지에만 적용할 수 있다는 단점이 있다. 또한 모든 스위치에 네트워크 전체의 트래픽 혼잡 정보를 저장하기 때문에 확장성 문제도 남아있다.

HULA[8]는 CONGA 의 문제점을 해결하고 더 짧은 Flow completion time (FCT)를 제공한다. 프로그래머블 스위치를 통해 주기적으로 probe 패킷을 보내 네트워크의 상태 정보를 파악한다. 또한 네트워크 전체의 트래픽 혼잡 정보를 각 스위치에 저장하지 않고 최적의 다음 홉 정보만을 저장한다. 이는 CONGA 에 비해 더 나은 낮은 flow completion time 을 갖으며, 모든 네트워크 토폴로지에 적용 가능하다는 장점이 있다.

CLOVE[9]는 네트워크의 트래픽 혼잡 정보만을 수집하는 CLOVE-ECN 과 네트워크의 정확한 경로 사용률을 수집하는 CLOVE-INT 두가지 방식으로 데이터를 수집 후 각각의 방식으로 경로의 가중치를 계산하고 이 두 가중치에 따라 weighted round-robin 방식으로 경로를 결정한다. 하지만 CLOVE 는 가상화된 스위치에서 동작하므로 가상화를 지원하지 않는 데이터 센터 네트워크에서는 CLOVE 를 사용할 수 없다는 한계점이 있다.

HULA 와 CLOVE 의 경우 네트워크 전체의 상태 정보를 파악하기 위해 추가적인 패킷을 보내기 때문에 네트워크 상태 정보 파악에 오버헤드가 있다는 단점이 있다. 본 논문에서는 P4 를 이용한 In-band 네트워크 텔레메트리 기법을 활용하여 추가적인 패킷을 보내지 않고도 실시간으로 네트워크 상태 정보를 파악하는 방법을 사용한다.

Ⅲ. 시스템 구조 제안

본 논문에서는 머신러닝으로 트래픽 패턴을 예측하여 경로의 가중치를 동적으로 변화시키는 방법을 제안한다. 제안하는 방법은 추가적인 패킷을 보내지 않고도 네트워크 전체의 상태 정보를 수집한다. 또한 머신러닝을 통해 대용량 트래픽이 흐를 것을 예측해 처리함으로써 기존의 중앙 집중화 방식에 비해 대용량 트래픽을 빠르게 처리할 수 있을 것으로 기대된다.

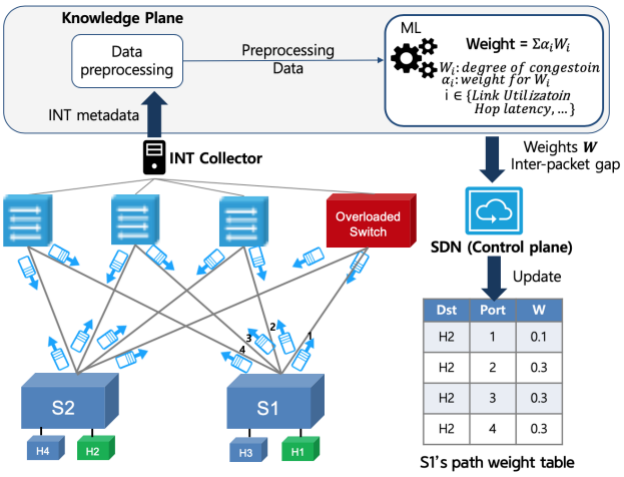


그림 1 동적 가중치 조정 로드밸런싱 예시

본 논문에서 제안하는 모델은 1) INT 메타데이터 수집 및 전처리 단계, 2) 경로 가중치 결정 단계 3) 경로 결정 단계로 구성되어 있다.

1) INT 메타데이터 수집 및 전처리 단계에서는 데이터 평면에서 얻은 네트워크 상태 정보를 INT Collector[15]를 통해 수집하는 단계이다. INT 메타데이터는 INT 를 사용하여 수집할 수 있는 네트워크 상태 정보로 각 패킷의 INT 헤더에 저장된다. 데이터 평면에서는 P4 를 이용해 INT 헤더를 패킷에 삽입하고, 관측되는 INT 메타데이터를 INT 헤더에 저장한다. 패킷의 INT 헤더 및 메타데이터는 INT Collector 로 전달되며, INT Collector 는 INT 헤더를 파싱하고 네트워크 상태 정보를 추출하여 저장한다. 저장된 데이터는 타임스탬프 정보를 활용하여 flowlet 단위의 통계 정보로 변환된다. 통계 정보의 종류로는 스위치의 버퍼 사용률, 패킷 전송속도, 링크 에러율, 링크 지연시간 및 대역폭 등이 있다. 통계 정보는 데이터 정제와 정규화와 같은 방법으로 전처리되어 머신러닝 모델로 전달된다.

2) 경로 가중치 결정 단계에서는 머신러닝을 통해 다음 네트워크 상태에 대한 경로 가중치를 계산한다. 머신러닝 모델은 네트워크 상태 통계 정보에 따라 링크 사용률, 홉

지연, 큐 사용량, 큐 혼잡도, 링크 당 플로우 수, 등의 측정치로 경로의 혼잡도를 계산하고 그 합으로 경로 가중치를 결정한다. 이때 머신러닝은 입력 받은 네트워크 상태 통계정보를 통해 네트워크의 트래픽 패턴을 예측한다. 그리고 예측한 트래픽 패턴에 따라 각 측정치에 의해 계산된 경로의 혼잡도를 조정한다. 다음 flowlet 을 나누는 기준이 되는 패킷 간의 IPG (Inter-packet Gap)도 위와 같이 트래픽 패턴에 따라 머신러닝으로 결정한다. 결정된 경로 가중치와 IPG 값은 SDN 컨트롤러에 전달된다.

3) 경로 결정 단계에서 SDN 컨트롤러는 2)에서 전달받은 경로 가중치로 각 스위치의 경로 가중치 테이블과 IPG 를 갱신한다. 이후 각 스위치는 들어온 패킷들을 경로 가중치 테이블에 따라 flowlet 단위로 weighted round-robin 방식으로 포워딩한다.

그림 1 은 본 논문에서 제안하는 로드밸런싱 방법의 예시이다. 각 패킷들은 In-band 네트워크 텔레메트리 기법을 통해 네트워크 상태 정보를 수집하여 INT Collector 로 전달한다. INT Collector 에서는 네트워크 상태 정보를 플로우 단위로 저장한다. 저장한 네트워크 상태 정보는 flowlet 단위의 통계 정보로 전처리 후 머신러닝 모델에 전달된다. 이를 바탕으로 각 경로의 가중치와 다음 flowlet 을 결정하는 패킷 간의 간격을 계산한다. 그림 1 의 경우 1 번 포트에 연결된 스위치가 과부하되었기 때문에 머신러닝에서는 해당 경로에 트래픽 정체가 발생할 것을 예측하고 1 번 포트로 나가는 경로의 가중치를 낮추게 된다. 마지막으로 각 경로의 가중치와 SDN 컨트롤러에 전달하여 각 스위치의 경로 가중치 테이블을 갱신한다. 스위치 S1 의 경우 1 번 포트의 경로 가중치는 다른 포트의 경로 가중치에 비해 낮아지게 된다. 이후에는 S1 의 경로 가중치 테이블에 따라 1 번 포트보다 다른 포트로 더 많은 패킷을 보내 부하를 분산시키게 된다.

Ⅳ. 결론

본 논문에서는 P4 를 이용한 In-band 네트워크 텔레메트리 기법을 사용해 네트워크 상태 정보를 수집하고, 이 정보를 통해 머신러닝으로 네트워크 변동 상황에 대해 빠른 반응성을 갖는 로드밸런싱 알고리즘을 제안하였다. 제안한 방법은 In-band 네트워크 텔레메트리 기법을 사용해 네트워크의 상태 정보를 수집할 때 발생하는 오버헤드를 최소화하면서 높은 네트워크 가시성을 확보한다. 또 머신러닝을 활용해 트래픽 패턴을 예측함으로써 네트워크 변동사항에 대해 빠르게 반응한다. 향후 연구로는 다양한 트래픽 패턴을 분석하고 이에 따라 경로의 가중치를 계산하는 머신러닝 모델을 개발할 계획이다. 또한 실제 실험 환경에서 기존의 다른 알고리즘과의 성능을 비교 분석할 계획이다.

ACKNOWLEDGMENT

이 논문은 2019 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (2018-0-00749, 인공지능 기반 가상 네트워크 관리기술 개발)

참 고 문 헌

[1] W. Xia, P. Zhao, Y. Wen and H. Xie, “A Survey on Data Center Networking(DCN): Infrastructure and Operations”, *IEEE Communication Surveys & Tutorials*, vol. 19, no. 1, pp. 640-656, 2017.

[2] J. Zhang, F. R. Yu, S. Wang, T. Huang, Z. Liu, and Y. Liu, “Load balancing in data center networks: A survey,” *IEEE Communicatoin Surveys & Tutorials*, vol. 20, no. 3, pp. 2324, - 2352, 2018.

[3] T. Benson, A. Anand, A. Akella, and M.Zhang, “Understanding data center traffic characteristics,” *Proceedings of the 1st ACM workshop on Research on enterprise networking - WREN* , 2009.

[4] M. Chiesa, G. Kindler, and M. Schapira, “Traffic Engineering with Equal-Cost-MultiPath: An Algorithmic Perspective,”*IEEE Conference on Computer Communications,* 2014

[5] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, “Hedera: Dynamic Flow Scheduling for Data Center Networks,” *NSDI,* 2010

[6] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, “Improving Datacenter Performance and Robustness with Multipath TCP,”*ACM SIGCOMM Computer Communication Review,* vol. 41, no. 4, p. 266,2011.

[7] M. Alizadeh *et al.*, “CONGA: distributed congestion-aware load balancing for datacenters,”*Proceedings of the 2014 ACM conference on SIGCOMM,*2014.

[8] N. Katta, M. Hira, C. Kim, A. Sivaraman, and J. Rexford, “HULA: Scalable Load Balancing Using Programmable Data Planes,”*Proceedings of the Symposium on SDN Research,*2016.

[9] N. Katta *et al.*, “Clove: Congestion-Aware Load Balancing at the Virtual Edge,” in *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies - CoNEXT* ,2017.

[10] P. Bosshart *et al.*, “P4: programming protocol-independent packet processors,” *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 87- 95, 2014.

[11] C. Kim, A. Sivaraman, N. Katta, A. Bas, A. Dixit, and L. J. Wobker, “In-band Network Telemetry via Programmable Dataplanes,” In *ACM SIGCOMM,* 2015.

[12] N.Viljoen *et al.*, “Machine learning based adaptive flow classification for optically interconnected data centers,” *18th International Conference on Transparent Optical Networks*, 2016

[13] S. Kandula, D. Katabi, S. Sinha, and A. Berger, “Dynamic load balancing without packet reordering,” *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 2, p. 51, 2007.

[14] N. Dukkipati and N. McKeown, “Why flow-completion time is the right metric for congestion control,” *ACM SIGCOMM Computer Communication Review*, vol. 36, no.1, p. 59, 2006.

[15] N. V. Tu, J. Hyun, G. Y. Kim, J.-H. Yoo, and J. W.-K. Hong, “INTCollector: A High-performance Collector for In-band Network Telemetry,” *IEEE*

Conference on Network and Service Management,
2018.

Modbus/TCP 프로토콜 기반 클러스터링 알고리즘 성능 평가

이민성, 심규석, 이민섭, 박준상*, 김명섭
고려대학교, *LG Electronics
{min0764, kusuk007, chenlima2, tmskim}@korea.ac.kr, *junsang.park@lge.com

Performance Evaluation of Modbus/TCP Protocol-based Clustering Algorithm

Min-Seong Lee, Kyu-Seok Shim, Min-Seob Lee, Jun-Sang Park*, Myung-Sup Kim
Korea Univ., *LG Electronics

요 약

산업 제어 기술이 발전하고 있고 자동화 기술이 발전함에 따라 네트워크 통신 환경이 중요하다. 네트워크 통신을 위한 통신 프로토콜이 필수적이며 그에 대한 연구가 필요하다. 특히 산업자동화의 증가에 따라 산업 프로토콜의 가치가 증가하고 이에 따른 산업 프로토콜 리버스 엔지니어링이 필요하다. 트래픽에 관련된 여러 가지 연구들이 진행이 되고 있고 다양한 방법으로 트래픽을 분석하고 있다. 본 논문은 프로토콜에 대한 두 가지 클러스터링 알고리즘의 성능을 분석하고 비교한다. 클러스터링 알고리즘의 성능을 분석 후 향후 프로토콜 리버스 엔지니어링 연구에 사용될 적합한 클러스터링 알고리즘을 제시한다. 본 실험에서는 산업 제어 기술 통신을 위한 프로토콜 리버스 엔지니어링에 K-means Clustering이 적합하지만 향후 더 다양한 Clustering 방법을 적용해 볼 필요가 있다.

Ⅰ. 서론

산업 제어 기술이 발전하고 있고 자동화 기술이 발전함에 따라 네트워크 통신 환경이 중요하다. 네트워크 통신을 위한 통신 프로토콜이 필수적이며 그에 대한 연구가 필요하다. 기존에 사용되던 프로토콜은 비트 전송의 신뢰도가 떨어졌고, 그것을 향상시킨 산업용 이더넷 통신이 주목을 받게 되었다. Modbus/TCP 프로토콜은 Modbus에서 한 단계 발전한 버전으로 TCP/IP를 활용하여 Modbus 메시지 전송을 구현한다[1]. 본 논문은 다양한 방법 중 Clustering 알고리즘을 통해 Modbus/TCP 프로토콜을 분석하였다.

Clustering 알고리즘은 주어진 데이터의 특성을 고려하여 군집을 형성하고 그 군집을 나타낼 수 있는 점을 찾는 방법이다. 이러한 Clustering 알고리즘을 Modbus/TCP 프로토콜에 적용하여 군집을 어떻게 형성하고 그 군집형성이 프로토콜의 특성에 알맞게 분류가 되었는지 확인하며 두 가지 Clustering 알고리즘의 성능을 분석하여 다양한 프로토콜들에 대한 적합한 Clustering 알고리즘을 제안하고자 한다.

Ⅱ. 본론

본 장에서는 Modbus/TCP 프로토콜의 구조에 대하여 정의하고 K-means Clustering 알고리즘과 Mean-Shift Clustering 알고리즘과 분석할 실험 데이터에 대해 언급한다.

2.1 Modbus/TCP 프로토콜 구조

본 절에서는 Modbus/TCP 프로토콜의 구조에 대하여 언급한다. Modbus/TCP 프로토콜은 MBAP(Modbus Application Protocol) 헤더와 함수 코드(Function Code), 그리고 함수 코드에 따른 Data로 이루어진다. MBAP 헤더에는 Transaction ID, Protocol ID, Length, Unit ID로 이루어져 있다. Transaction ID는 Query 및 Response 한 쌍의 작업으로 구분하기 위해 사용되는 번호이며 마스터에 의해 설정된다. 최초 0x0000값부터 통신 시작 시 1씩 증가시킨다. Protocol ID는 Protocol의 ID를 나타내며 0x0000으로 고정 값이다. Length는 Length 필드 위치에서 프레임 마지막까지의 길이를 나타내며 Unit ID에서부터 Data 끝까지의 Byte 수이다.Unit ID는 TCP/IP가 아닌 다른 통신선로의 연결되어있는 서버를 구분하며 TCP 포트는 0x01로 고정이다. 함수 코드는 Modbus 프로토콜에서 제공하는 명령어 집합 코드로서 Memory에 값을 읽어오거나 쓸 수 있는 서비스이다. 본 연구실에서 수집한 Modbus/TCP 트래픽의 함수 코드는 90으로 고정이다. Data는 함수 코드에 따라 구조가 조금씩 달라진다. 본 논문에서 다룰 Clustering 알고리즘의 데이터는 Modbus/TCP의 Data 부분이다.

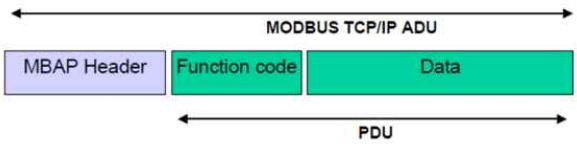


그림 1. Modbus/TCP 프로토콜 구조

표 1. MBAP 헤더

항목	길이
Transaction ID	2 bytes
Protocol ID	2 bytes
Length	2 bytes
Unit ID	1 bytes

2.2 Clustering Algorithm

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

본 절에서는 성능평가에 사용되는 두 가지 Clustering 알고리즘에 대해 언급한다. 선정된 두 가지 Clustering 알고리즘은 K-means와 Mean-Shift 알고리즘이다.

K-means 알고리즘은 주어진 데이터들 다시의 거리 혹은 유사성을 이용하여 K개의 클러스터로 군집시켜주는 알고리즘으로 K는 구분하고자 하는 군집의 개수를 의미한다[2]. 이는 사용자가 임의로 원하는 군집의 개수를 정할 수 있다는 것을 의미한다. K-means 알고리즘은 사전 정보가 필요 없이 거리만 가지고 군집화가 가능하지만 군집의 개수를 사용자가 정해야한다는 단점이 있다. 실험에서는 elbow기법을 통해 군집의 개수를 정의한다.

Mean-Shift 알고리즘은 밀도기반 Clustering 알고리즘으로서 데이터의 무게중심을 찾아가는 방식이다[3]. 자신의 주변에서 원을 그리며 가장 데이터가 밀집된 방향으로 이동하며 수렴할 때 그 데이터가 군집의 중심이 된다. Mean-Shift 알고리즘은 평균이동이 자동으로 각 군집의 중심 데이터를 찾기 때문에 군집의 개수를 선택할 필요가 없다. 하지만 원의 크기(bandwidth)에 따라 군집의 개수가 달라진다. 실험에서는 Bandwidth의 quantile값을 권장값인 0.3으로 설정하여 실험한다.

2.3 실험 Data

본 절에서는 실험에 사용된 Modbus/TCP에서 Data 부분이다. Data는 함수 코드에 따라 달라지는데 기본적으로 Start Address, Length, Byte Count, Data의 형태를 가진다.

표 2. DATA

항목(길이)	설명
Start Address[2 Bytes]	접근하려는 메모리의 시작번지
Length[2 Bytes]	시작번지부터 값을 읽거나 쓸 길이
Byte Count[1 Bytes]	Request, Response에 따른 메모리 Data byte 수
DATA[N Bytes]	Request, Response에 따른 메모리 Data의 값

Ⅲ. 실험 및 결과

본 장에서는 선정된 클러스터 알고리즘을 사용해 실험을 진행한다. 실험과정은 실험에 사용할 Modbus/TCP의 Data를 추출한 후에 Size별로 구분 후 각 Size별로 두 개의 Clustering 알고리즘을 진행한다.

본 연구실에서 수집한 약 22개의 트래픽 셋의 Modbus/TCP의 Data를 추출하여 길이별로 구분을 한다. 총 7079개의 Data들 중 Size별 Clustering을 할 필요가 없는 개수가 적은 표본들을 제외하고 실험을 진행한다.

표 3. 실험 데이터

Data Length	개수	Data Length	개수
2	1327	22	35
3	599	28	96
4	78	49	42
6	684	68	705
8	69	105	226
9	210	161	134
11	1027	1019	99
12	365	1020	615
14	162		

본 실험에서 K-means와 Mean-Shift 방식의 Clustering의 군집화 개수는 크게 차이가 나지 않지만 K-means에서 군집도를 보다 더 정교하게 나타낸다는 것을 나타낸다. 실험결과의 한 예시중 Length가 1020인 Data Clustering을 실험한 결과, K-means의 경우 타입을 구분시 다음 결과와 같이 T로 가득찬 메시지와 00부터 FF까지 순차적으로 증가하는 메시지를 구분한다. 그러나, Mean-Shift의 경우 두 개의 타입을 구분해내지 못하기 때문에 현재 K-means가 산업 제어 기술 통신을 위한 프로토콜 리버스 엔지니어링 기술에 적합하다. 하지만, 향후 UPGMA와 Needleman-Wunsch등 더 다양한 Clustering 방법을 적용해 볼 필요가 있다.

표 4. 실험 결과

Length	K-means cluster	Mean-Shift cluster
2	4	4
3	8	8
4	5	4
6	4	6
8	3	2
9	5	4
11	4	8
12	4	4
14	7	2
22	5	4
28	4	2
49	6	4
68	6	6
105	7	4
161	4	4
1019	3	3
1020	4	1

표 5. K-means Type(Length : 1020)

Method	Type	
K-means	Type 0	"T,...T"
	Type 1	"00,01,02,...,FF"

Ⅳ. 결론 및 향후 연구

본 논문은 산업 제어 기술 통신을 위한 통신 프로토콜에 대한 두 가지의 Clustering 알고리즘을 실험하였다. 실험한 결과 향후 연구에 사용될 프로토콜 리버스 엔지니어링에는 K-means Clustering이 적합하다. 하지만 더 다양한 Clustering 방법을 적용해 볼 필요가 있으다. 또한, 산업 제어 기술 통신 프로토콜뿐만 아니라 사용되고 있는 다양한 프로토콜에 대한 Clustering 연구도 필요하다.

향후 연구로는 Netzob에서 사용중인 두가지 Clustering 방식인 UPGMA와 Needleman-Wunsch 알고리즘을 통한 실험을 진행 할 예정이며 상용 프로토콜에서 어떤 Clustering 알고리즘이 적합한 결과를 도출할 것인지 실험할 계획이다.

참 고 문 헌

[1] Denton, G., Karpisek, F., Breitinger, F., & Baggili, I."Leveraging the SRTP protocol for over-the-network memory acquisition of a GE Fanuc Series 90-30". Digital Investigation, 22, 2017, S26-S38.

[2] Jain, A.K, "Data clustering: 50 years beyomd K-means.", Pattern recognition letters, 2010, pp.651-666

[3] Comaniciu, Dorin, and Peter Meer. "Mean shift: A robust approach toward feature space analysis." IEEE Transactions on Pattern Analysis & Machine Intelligence 5, 2002, pp.603-619.

[4] G. Bossert, "Exploiting semantic for the automatic reverse engineering of communication protocols," Ph.D. dissertation, Univ. Gif-sur-Yvette, Rennes, France, Dec. 2014.

차세대 국제 연구망 구축을 위한 네트워크 인프라 설계 요건에 관한 연구

조부승[1,2], 장민석[1,*]

[1]한국과학기술정보연구원, [2]과학기술연합대학원대학교

{bscho, msjang}@kisti.re.kr

*교신저자

A Study on the Requirement for the Planning and Design of the Next Generation International Research Network

Buseung Cho, Minseok Jang

Korea Institute of Science and Technology Information, University of Science and Technology

요 약

최근 전 세계의 국가 연구망은 데이터집약형과학(Data Intensive Science) 및 4차 산업혁명시대에 요구하는 대용량 빅데이터를 효율적으로 전송하기 위한 국가 연구망 인프라에 대한 고도화를 추진하고 있다. 이를 위해 국가 연구망 이용 집단의 요구사항, 최신 네트워크 기술동향, 이용 집단의 서비스 요구사항 등을 반영한 네트워크 인프라의 설계는 필수적이다. 데이터집약형 과학분야의 글로벌 협업을 가능하게 하기 위해서는 대용량 데이터 전송이 필수적이며, 이를 효율적으로 지원하기 위한 lightpath를 비롯한 다양한 계층별 VPN 서비스를 비롯하여 이용 그룹의 응용의 특성에 부합하는 네트워크 서비스가 필요하다. 또한 국제 연구망의 경우, 해저케이블의 활용은 필수적이며, 이 경우 아직은 100기가 람다에 대한 임차비용이 고가인 상황에서 실제 연구망을 이용하는 이용자의 요구되는 망자원에 대한 수요 분석한 후 향후 3년 혹은 5년 정도의 네트워크 업그레이드 주기를 고려하여 수요에 대응 가능한 비용대비 효율적인 국제 연구망의 구축을 위한 자원 확보 전략이 필요하다.

Ⅰ. 서 론

최근 전 세계의 국가 연구망은 데이터집약형과학(Data Intensive Science) 및 4차 산업혁명시대에 요구하는 대용량 빅데이터를 효율적으로 전송하기 위한 국가 연구망 인프라에 대한 고도화를 추진하고 있다. 일반적으로 국가 연구망은 짧게는 3년 혹은 5년 주기로 네트워크 인프라 고도화를 추진하고 있으며, 이를 위해 이용 집단의 요구사항, 최신 네트워크 기술동향, 이용 집단의 서비스 요구사항 등을 반영한 네트워크 인프라의 설계는 필수적이다. 특히 최근 유럽최대입자물리연구소(CERN)의 강입자가속기(Large Hadron Collider)를 비롯하여, 초대형 전파망원경(Square Kilometre Array) 등의 대형 연구 장비에서 생산된 대용량 실험데이터를 이용하여 전 세계 연구자들이 협업연구를 원활히 수행하기 위한 국제 연구망은 더욱 중요해지고 있다.

본 논문에서는 차세대 국제 연구망 고도화를 위한 네트워크 인프라 설계를 위해 위에서 언급한 다양한 설계 요건을 분석하고, 이를 통한 설계 요건을 정의하고자 한다. 이를 위해, 국가 연구망을 활용하는 이용집단에 대한 네트워크 자원(대역폭), 서비스 요구사항 등을 조사 및 분석함으로써, 네트워크 인프라의 양적인 자원의 확보 수준은 물론 서비스 요구사항 분석을 통한 최신 네트워킹 기술의 적용유무를 결정할 수 있다. 또한 글로벌 협업 연구를 원활히 지원하기 위해 국제 연구망은 일반적인 상용 캐리어 사업자와는 달리 자연스러운 연구협업을 보장하기 위한 다양한 성능 요건을 만족해야 하는 등 다양한 연구망의 특수성을 고려하여 글로벌 연구 플랫폼으로써의 국제 연구망의 목적에 부합하는 설계가 필요하다.

Ⅱ. 관련 동향 및 연구

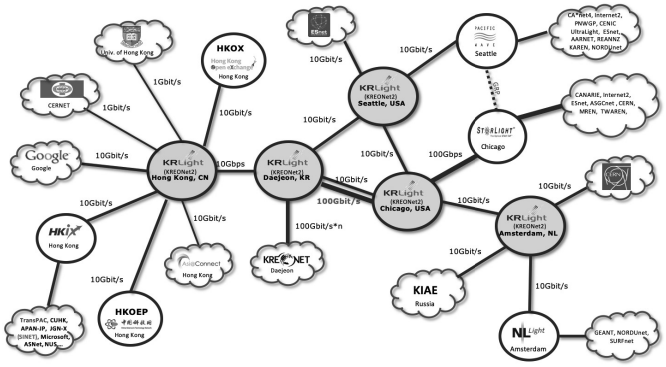
100기가 혹은 400기가 람다 전송기술의 개발, 비용대비 효율적인 OTN 및 캐리어이더넷 기술의 보급 및 확대[1], 회선 장애에 대한 생존성을 보장하고 동적인 대역폭 자원의 할당[2] 및 개방형 ROADM 장비 개발[3] 등 광전송기술의 발전과 더불어 데이터집약형과학(Data Intensive Science) 및 4차 산업혁명시대에 필요시 되고 있는 대용량 데이터를 전송하는 국가 연구망의 네트워크 인프라의 고도화가 지속적으로 진행 중이다. 특히 미국, 유럽 등에서는 IRU 기반의 광케이블에 대한 장기 임대, 테라급 광전송 및 데이터처리 장비의 도입, 이를 통한 국가 연구망 서비스의 고도화를 도모하고 있다.

Ⅲ. 본론

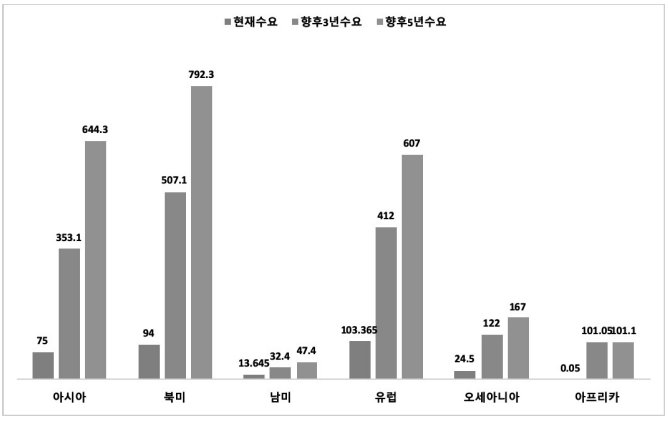
차세대 국제 연구망 인프라 설계를 위해, 가장 먼저 국제 연구망에 대한 실제 이용 집단 혹은 이용자의 대역폭으로 대표되는 국제 연구망 인프라 자원 및 서비스에 대한 수요를 조사하고 이를 분석함으로써 국제 연구망 인프라에 대한 설계 요건을 살펴본다. 또한 프러덕션 수준의 국제 연구망 서비스를 위한 망의 서비스 가용율 향상, 글로벌 협업 연구에서 요구하는 최대 전송 성능 보장 서비스, 전송성능 대비 투자비용의 효율성 등 국제 연구망 인프라 고도화 측면에서의 설계 조건 또한 알아본다.

국가 연구망의 국제 연구망 인프라인 글로벌과학기술협업연구망(GLORIAD)를 실제 이용하고 있는 고에너지물리, 천문우주, 게놈/바이오, 건설건축, 미래네트워크, 기상기후, 빅데이터융합 등으로 분류되는 7개 글로벌 첨단협업 연구 분야의 국내 22개 핵심연구그룹에 대하

여, 2018년 7월 7일부터 7월 18일까지 조사하였다.[4] 그 결과 국제 연구망 대역폭에 대한 이용그룹의 향후 3년 내 수요는 현재 확보 수준(130Gbps)의 약 11배 수준인 1,426.6Gbps이며, 향후 5년 내 수요는 현재 확보 수준의 약 174배 수준인 2,258Gbps로 나타났다. 특히 아시아권(홍콩 경유)은 향후 3년 내 353.1Gbps가 요구되어 현재 확보 수준(10Gbps)의 35배 이상을 요구하고 있으며, 북미/남미/유럽/오세아니아권은 향후 3년 내 1,073.5Gbps가 요구되어 현재 확보 수준(110Gbps) 대비 약 10배 수준으로 요구하고 있다. 세부적으로 북미 서부권과 남미/오세아니아의 경우 향후 3년 내 424.4Gbps, 즉 현재 확보 수준(10Gbps 시애틀 경유)의 42배 이상을 요구하고 있으며, 북미 북중권과 유럽/아프리카의 경우 향후 3년 내 809Gbps, 즉 현재 확보 수준(100Gbps 시카고 경유)의 경우 8배 이상을 요구하고 있다. 또한 2015년 조사된 이용그룹에 대한 대역폭 자원 수요조사 결과, 현재 확보 수준의 2배 정도인 331.5Gbps인 점을 감안하면, 향후 요구되는 대역폭 자원의 수준은 과거와 비교해 볼 때, 대역폭 자원에 대한 수요가 폭발적으로 증대되고 있다고 볼 수 있다.



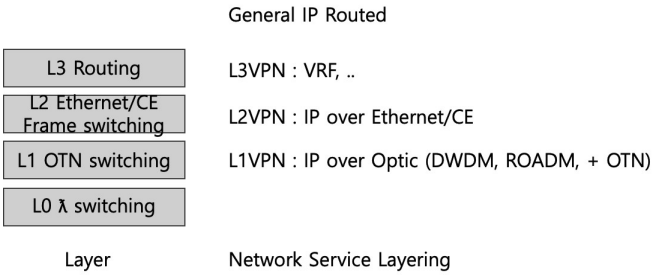
<그림 1> 글로벌과학기술협업연구망 자원확보 및 연동 현황(구성도)



<그림 2> 국제 연구망 자원에 대한 수요

국제 연구망의 대역폭 자원에 대한 요구사항을 인프라 설계에 반영 시 고려되어야 할 요소는 협업연구 시에 활용되는 응용의 특성은 최대 1페타급 데이터를 1주일 내 전송(초당 92기가급 전송) 혹은 8K 비압축 초고해상도 멀티미디어 전송(초당 40기가급 전송) 등으로 초당 100기가급 전송이 가능한 네트워크 환경이 필요하며, 특정 시점에서는 종단간 100기가급 전송이 가능해야 한다. 또한 추가적인 고려사항으로 해저케이블에 대한 임차비용으로 한국에서 아시아(홍콩), 그리고 북미(시애틀/시카고)로의 인프라 구축에 있어 해저케이블 임차는 필수적이며, 최근 아시아태평양 구간의 경우 10Gbps 회선의 임차는 3년 전에 비해 50%수준으로 하락했으며, 100Gbps 회선의 임차비용은 10Gbps 회선임차비용의 5배 수준으로 나타나고 있어[5], 이를 고려한 비용대비 자원 확보율에 대한 고려가 필요하다.

다음으로 이용그룹이 요구하는 서비스 중 가장 중요시 되는 서비스는 초고성능 데이터 전송 서비스로 이는 연구망의 이용하는 이용그룹의 특성상 일반적으로 elephant flow로 대표되는 대용량 빅데이터를 빠르게 전달할 필요가 있기 때문이다. 또한 최근 데이터의 공유와 함께 데이터 저장 및 공유 서비스를 비롯하여, 클라우드 서비스(가상머신 및 컨테이너 서비스), 보안서비스, IoT 센서 네트워크 서비스 등으로 나타났다. 이 중 네트워크 인프라와 밀접한 관련이 있는 부분은 초고성능 데이터 전송 서비스로써 이는 WAN 구간인 국제 연구망에서의 성능보장형 VPN (Virtual Private Network) 서비스가 핵심적으로 요구되고 있음을 확인할 수 있다. 대표적으로 이를 구현하기 위해 광패스(Lightpath)로 대표되는 L1VPN 그리고 캐리어이더넷 기술을 활용한 L2VPN, 마지막으로 L3VPN 등이 요구되며, 실제 고에너지물리분야 WLCG (Worldwide LHC Computing Grid) CERN LHC 데이터센터간의 네트워크인 LHCOPN (LHC Optical Private Network) 및 LHCONE (LHC Open Network Environment)으로 쉽게 확인할 수 있다. 특히, 사용자 요구 기반 대역폭 보장형 서비스(Bandwidth on Demand)의 구현을 위한 계층별 최적의 전송 및 데이터 장비에 대한 고려가 요구된다. 최근 PTN (Packet Transport Network) 장비로 구분되는 패킷광전송장비를 통한 계층간 통합, 람다당 100Gbps/200Gbps 전송기술의 보편화 및 400Gbps 전송기술의 상용화 등 인터페이스당 전송용량에 대한 전송성능 대비 투자비용, OTN으로 대표되는 서비스 투명성 보장 및 서비스 가용율 확보를 위한 고가용성 보장 등에 대해서도 추가 고려되어야 한다.



<그림 3> 네트워크 서비스 계층화

최근 기존의 전송장비 벤더의 폐쇄성에서 벗어나 전송장비의 유연성 및 개방성을 강조한 개방형 전송 시스템(Open Line System)에 대한 고려가 필요할 수 있으나, 본 연구에서는 네트워크 인프라 구축/활용 주기를 3년으로 하고 있어, 장기적인 투자가 필요한 개방형 전송 시스템에 대한 고려는 하지 않는 것으로 한다. 또한 본 논문에서는 국가 연구망의 국제 WAN 구성에 대한 설계 요건에 대한 고려로 SDN/NFV로 대표되는 네트워크의 가상화 및 programmability를 기반으로 하는 기술에 대해서는 WAN 시장에서는 Overlay SD-WAN 솔루션이 대부분이지만, 최근 인공지능, 운영자동화 등이 부각되고 있으며 이는 지원하는 Underlay SD-WAN에 대한 고려가 필요하다. 여전히 전 세계 대부분의 연구망에서 안정성, 보안성, 호환성 등에 대한 문제가 지적되고 있고, 아직 프로덕션 수준의 Underlay SD-WAN 인프라 구축이 많이 진행되고 있지 않다. 현재 기존의 국가 연구망(KREONET)의 SD-WAN(KREONET-S)에서는 일반적인 스위칭과 라우팅으로 동작하는 데이터처리 영역과는 별도의 SD-WAN 영역을 구축하여 서비스한다는 점에서 L0/L1 스위칭 기반의 Lightpath (point-to-point 회선)를 SD-WAN 장비 간 제공하는 것으로 구성한다.

Ⅲ. 결론

본 논문에서는 차세대 국가 연구망에 대한 네트워크 인프라 설계 시 고려되어야 할 설계 요건들에 대해 알아보았다. 최근 국가 연구망은 데이터집약형 과학분야의 글로벌 협업을 가능하게 하기 위한 대용량 데이터 전송이 필수적이며, 이를 효율적으로 지원하기 위한 lightpath를 비롯한 다양한 계층별 VPN 서비스를 비롯하여 이용 그룹의 응용의 특성에 부합하는 네트워크 서비스가 필요하다. 또한 국제 연구망의 경우, 해저케이블의 활용은 필수적이며, 이 경우 아직은 100기가 람다에 대한 임차비용이 고가인 상황에서 실제 연구망을 이용하는 이용자의 요구되는 망자원에 대한 수요 분석한 후 향후 3년 혹은 5년 정도의 네트워크 업그레이드 주기를 고려하여 수요에 대응 가능한 비용대비 효율적인 국제 연구망의 구축을 위한 자원 확보 전략이 필요하다.

ACKNOWLEDGMENT

본 연구는 과학기술정보통신부 및 한국과학기술정보연구원의 주요사업의 일환으로 수행하였음. [과제명: 글로벌 협업연구 지원 국가 과학기술연구망 구축 및 서비스]

참 고 문 헌

[1] Mohcene Mezhoudi, Ying (Emily) Hu, “Optical Backbone Network Evolution: Design, Optimization and Evaluation of NG-OTN”, 2010

[2] Kwok Shing Ho and Kwok Wai Cheung, “Generalized Survivable Network”, IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 15, NO. 4, AUGUST 2007

[3] Geoff Bennett, “Open Line Systems and Open ROADM: How Open Is Your Line System?”, TNC2018

[4] IHS Market, “국가과학기술연구망 수요조사 및 가치분석”, 2018

[5] Alan Mauldin, “Submarine Cable and Capacity Pricing Trends in Asia-Pacific”. APRICOT 2018

고신뢰성 5G 코어망을 위한 네트워크 자동화 기법 연구

이재욱, 고한얼, 이호찬, 백상헌
고려대학교

{iioiioiio123, heko, ghcks1000, shpack} @korea.ac.kr

A Study on the network automation for supporting high reliability to 5G core network

Jaewook Lee, Haneul Ko, Hochan Lee, Sangheon Pack
Korea Univ.

요 약

5G 의 대표 서비스들인 자율주행, 가상현실 및 증강현실과 같은 서비스들은 초저지연의 전송시간과 높은 신뢰성을 요구한다. 본 논문에서는 고신뢰성 지원을 위한 3GPP 표준화 동향과 기계학습 및 인공지능을 통한 네트워크 자동화를 위한 3GPP 표준화 동향을 분석한다. 끝으로, 5G 코어 네트워크에서 고신뢰성을 지원하기 위한 네트워크 자동화 프레임워크와 네트워크 자동화를 위한 요소 기법들을 분석한다.

Ⅰ. 서 론

자율주행, 가상현실, 증강현실과 같은 5G 의 대표 서비스들은 기존의 서비스들보다 낮은 지연시간과 높은 신뢰성을 요구한다. 이러한 5G 서비스 요구사항을 지원하기 위해, 3rd Generation Partnership Project (3GPP)에서 5G 구조와 기술을 표준화하고 있다. 대표적으로 고신뢰성과 초 저지연성을 요구하는 서비스인 Ultra-Reliable Low-latency Communication (URLLC) 서비스를 위한 표준화 작업이 2018 년 4 월부터 스터디 아이템으로써 진행되었으며 [1], 현재는 논의된 부분 중 주요한 부분들이 표준화 스펙에 포함되어 표준화가 진행되고 있다 [2].

5G 코어에서 고신뢰성을 지원하기 위해 중복전송을 허용하는 것을 채택하였다. 그림 1 과 같이 Host A 와 Host B 간의 서로 다른 경로에 PUD 세션들을 설정하여 같은 데이터를 전송하는 방식으로 신뢰성을 높일 수 있다. 3GPP 에서는 중복전송기법에 대한 표준화는 진행하지 않으며, 중복 경로의 패킷/프레임 복제 및 제거는 IEEE TSN (Time Sensitive Networking) FRER (Frame Replication and Elimination for Reliability)와 같은 상위 계층 프로토콜에 의해 관리하는 것으로 정의하였다. 그림 1 에서의 gNB 와 UPF 는 5G 에서의 기지국과 코어망에서의 데이터 평면을 의미한다. 패킷 복제 및 제거를 담당할 수 있는 Redundancy Handling Function (RHF)은 네트워크 망 내에서도 존재할 수 있으며, 본 논문에서는 UPF 가 해당 기능을 지원할 수 있다고 가정한다.

3GPP 에서는 URLLC 서비스와 같이 5G 서비스를 위한 구조 및 기술을 위한 표준화 작업과 함께, 네트워크 자동화를 위한 표준화도 진행 중에 있다 [3-4]. 3GPP 에서는 네트워크 자동화를 위해 network data analytics function (NWDAF)을 3GPP 엔티티로 정의하였으며, 해당 기능은 네트워크 데이터를 수집하고 분석하는 역할을 한다. 그림 2 는 NWDAF 를 통한 네트워크 자동화를 위한 프레임워크를 나타낸다. NWDAF 는 5G 의 다른 네트워크 기능 (NF)와 Edge

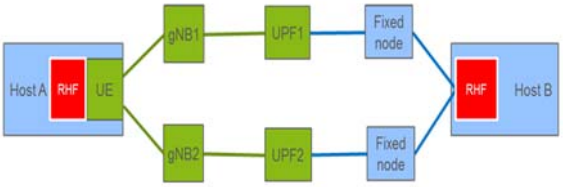


그림 1. 단일 디바이스에서의 High level 아키텍처

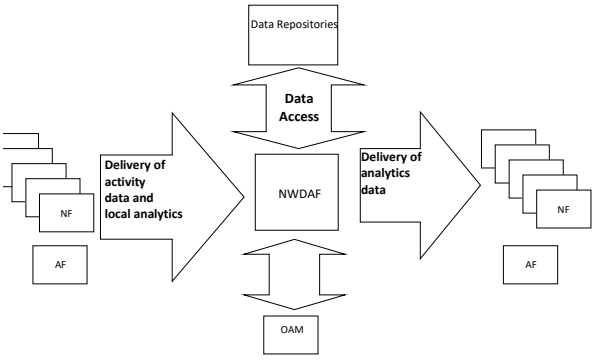


그림 2. 5G 네트워크 자동화 프레임워크

computing 을 관리하는 application function (AF) 그리고 네트워크 사업자 (OAM)들로부터 데이터를 수집하고, 수집된 데이터를 관리자의 의도에 맞게 분석한다. 또한 분석된 데이터를 NF, AF 또는 관리자에게 제공하여 네트워크를 관리한다.

본 논문에서는 고신뢰성 5G 코어망을 위한 네트워크 자동화 기법에 대해 제안한다. 본론에서는 네트워크 자동화를 위한 프레임 워크와 후보 요소 기술을 소개하고, 결론에서는 향후 연구방향을 제안한다.

Ⅱ. 본론

그림 3 은 제안하는 네트워크 자동화 프레임워크를 나타낸다. Session Management Function (SMF)는 5G 코어망의 데이터 평면인 UPF 들을 관리하는 기능으로써

네트워크 상태 정보를 수집할 수 있으며, 네트워크 상의 경로를 설정할 수 있다. 본 논문에서는 SMF 가 수집한 네트워크 상태 정보 (링크 손실율 및 지연속도 등)는 데이터 저장소 (Unstructured Data Storage Function (UDSF) 또는 Structured Data Storage Function (SDSF))에 저장된다. 즉, SMF 가 주기적으로 UPF 로부터 네트워크 상태정보를 수집하여 데이터 저장소에 저장하고, NWDAF 는 UE 의 서비스 요구사항을 충족하기 위해 저장소의 정보를 분석하여 중복전송 여부와 중복전송 경로를 결정한다. 결정된 중복전송 경로에 대한 결과를 SMF 에게 전달하고, SMF 는 받은 결과에 따라 중복전송 경로를 구축하고 중복전송을 수행한다.

데이터 저장소에 정보가 많이 존재하고, 최적의 결정을 위한 최적화 툴의 입력으로 용이한 데이터로 가공되는 경우를 가정하여, 그림 4 에서 실선 화살표와 같이 모델링 기반의 네트워크 자동화 기법이 적용 가능하다. 해당 기법은 우선 네트워크를 모델링하고, 모델링 된 정보를 기반으로 최적화 기법을 통해 관리자의 의도에 맞게 최적의 네트워크 자동화를 수행한다. 최근 심층학습 (Deep learning)을 통해 높은 정확도로 네트워크 모델링하는 연구가 활발히 진행되고 있다 [5][6]. 따라서, NWDAF 는 추론 (inference)에 우수한 성능을 보이는 Variational AutoEncoder (VAE) [5] 혹은 그래프 데이터에 우수한 성능을 보이는 Graph Neural Networks (GNN) [6]를 통해 네트워크를 높은 정확도로 모델링 가능하다. 학습시킨 인공지능망을 통해 관리자의 의도를 만족하는 최적의 중복전송 경로를 결정하기 위해 NWDAF 의 Optimizer 를 정의해야 한다. 예를 들어 네트워크 관리자가 서비스가 요구하는 신뢰성을 보장하면서, 중복 전송되는 패킷 양의 최소화를 요구한다면, Optimizer 는 integer linear programing (ILP)와 같은 최적화 기법을 통해 정의된다. 반면에 PUD 세션 변경에 따른 UPF 와 SMF 간의 컨트롤 메시지 교환을 최소화하는 경우, 일시적인 상태에서의 최적의 해를 제공하는 ILP 기법보다 지금 시점으로부터 이후 시점까지를 고려하여 최적의 해를 찾는 Constrained Markov Decision Process (CMDP)와 같은 최적화 기법을 통해 모델링 가능하다.

모델링 기반의 네트워크 자동화 기법은 네트워크 모델링의 정확성이 자동화 성능에 큰 영향을 끼친다. 즉, 데이터 저장소에 데이터 양이 충분치 않거나, 데이터 모델링이 힘든 상황에서는 NWDAF 가 경험을 통해 네트워크를 배워가는 경험기반의 네트워크 자동화 기법 (그림 4 에서 점선 화살표)이 적용 가능하다. [7]와 같이 심층강화학습기반으로 동작하는 경우에는, NWDAF 는 여러 상황에서 중복 전송 기법을 결정하고 SMF 는 결정에 대한 피드백을 NWDAF 에게 제공해준다. 해당 피드백을 통해 NWDAF 는 특정상황에서 어떤 결정이 최적인지를 학습하게 됨으로써, 최적의 네트워크 자동화를 수행할 수 있다.

Ⅲ. 결론

본 논문에서는 고신뢰성 지원을 위한 3GPP 표준화 동향과 기계학습 및 인공지능을 통한 네트워크 자동화를 위한 3GPP 표준화 동향을 분석하였다. 끝으로, 5G 코어 네트워크에서 고신뢰성을 지원하기 위한 네트워크 자동화 프레임워크와 네트워크 자동화를 위한 요소 기법들을 분석하였다. 향후 연구로는 최적화 모델링을 통해 본 기법들을 고도화하고, 고 신뢰성을 위한 네트워크 자동화를 위해 데이터 평면에 필요한 기능

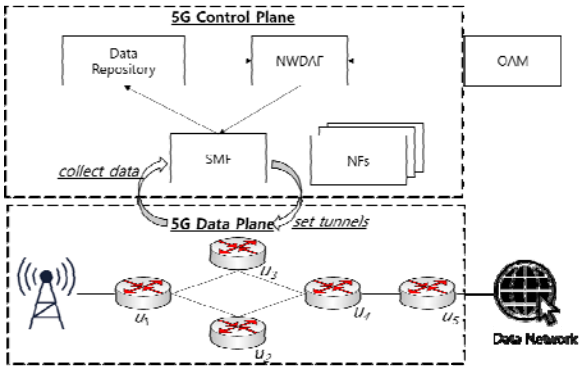


그림 3. 고신뢰성 지원을 위한 네트워크 자동화 프레임워크

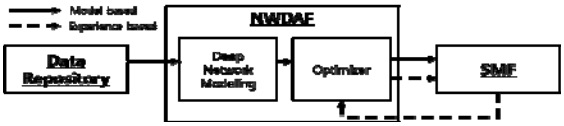


그림 4. 네트워크 자동화 동작과정

(패킷 손실률 측정 및 보고 기능, RHF 기능 등) 들을 구현하여 테스트베드를 구축할 예정이다.

ACKNOWLEDGMENT

이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(No.2015-0-00575, 글로벌 SDN/NFV 공개소프트웨어 핵심 모듈/기능 개발)과 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터지원사업의 연구결과로 수행되었음. (IITP-2019-2017-0-01633)

참 고 문 헌

[1] 3GPP TR 23.725 v2.0.0, "Study on enhancement of Ultra-Reliable Low-Latency Communication (URLLC) support in the 5G Core network (5GC)," 2018.12.

[2] 3GPP TS 23.501 v16.0.2, "System architecture for the 5G System (5GS), Stage 2," 2019.04.

[3] 3GPP TR 23.791 v2.0.0, "Study of enablers for Network Automation for 5G," 2018.12.

[4] 3GPP TS 23.288 v0.4.0, "Architecture enhancements for 5G System (5GS) to support network data analytics services," 2019.04.

[5] S. Xiao, *et al.*, "Deep-Q: Traffic-driven QoS Inference using Deep Generative Network," in *Proc. ACM SIGCOMM 2018 Workshop on Network Meets AI & ML*, Budapest, Hungary, August 2018.

[6] Z. Xu, *et al.*, "Unveiling the potential of Graph Neural Networks for network modeling and optimization in SDN," *in Proc. ACM Symposium on SDN Research (SOSR) 2019,* CA, USA, April 2019.

[7] Z. Xu, *et al.*, "Experience-driven Networking A Deep Reinforcement Learning based Approach," *in Proc. IEEE International Conference on Computer Communications (INFOCOM) 2018,* Hawaii, USA, April 2018.

De-anonymizing Bitcoin through mapping Transactions and public-keys to nodes

Meryam Essaid, Sejin Park, Hongteak Ju
Department of Computer Engineering, Keimyung University

meryamesd@stu.kmu.ac.kr, {baksejin,juht@kmu.ac.kr}

Abstract

The Bitcoin system is a pseudo-anonymous payment system and cryptocurrency that can separate the virtual Bitcoin users identities from any real-world identities. In that context, a successful breach of the virtual and physical divide represents a significant flaw in the Bitcoin system. In this work, we developed a methodology on how to extract information about the real-world users behind Bitcoin transactions from the circulating data in the network. We analyze publicly available data about the cryptocurrency. In particular, we focus on determining information about a Bitcoin user's physical location by examining users' spending habits.

Ⅰ. Introduction

Bitcoin [1] is a Peer-to-Peer Electronic Cash System with no central authority or units. There is no requirement for a trusted third-party when making transactions. Suppose User A wishes to send a number of BTC to User B. User A uses a Bitcoin client to join the Bitcoin network and makes a public declaration stating the identities he owns (which can be verified using public-key cryptography), the created transaction will contain the amount of BTC to be sent, the inputs (which previously had a number of BTCs assigned to one or more other identities) and a number of outputs, which at least one of them is controlled by user B. The participants of the P2P network form a collective consensus regarding the validity of this transaction by appending it to the public history of previously agreed-upon transactions (the longest ledger). This process is known as mining, involves the repeated computation of a cryptographic hash function so that the digest of the transactions, along with the transactions, and an arbitrary nonce has a specific form (block). This process is designed to require considerable computational effort, from which the security of the Bitcoin mechanism is derived.

Bitcoin is a decentralized digital currency whose transactions are recorded in a common ledger, so-called blockchain. Due to the anonymity and lack of law enforcement, Bitcoin has been misused in darknet markets [16] which deal with illegal products, money laundering, gambling. Therefore from the security forensics aspect [17-19], it is demanded to establish an approach to map between users' spending habits and their real-world identities. Mapping users' pseudonyms to their real users' identities are essential to discern the different economic activities conducted by the pseudonymous actors.

In this work, we developed a methodology on how to extract information about the real-world users behind Bitcoin transactions from the circulating data in the network. We analyze publicly available data about the cryptocurrency. In particular, we focus on determining information about a Bitcoin user's physical location by examining users' spending habits.

The rest of the paper is organized as follows: Section 2 introduces the Bitcoin system. The used Bitcoin De-anonymizing methodology is described in Section 3. Section 4 and 5 respectively explain the topology discovery mechanism and the used clustering system. Section 6 underline some of the strengths and weaknesses of the proposed solution. Section 7 points to related work. Finally, Section 8 discusses the conclusion and future work.

II. Bitcoin System

There are three main features of the Bitcoin system that are of our particular interest. Firstly, the entire history of Bitcoin transactions is publicly available. The second feature of interest is that a transaction can have multiple inputs and multiple outputs. A transaction frequently has either a single input from a previous larger transaction or multiple inputs from previous smaller transactions. Also, a transaction frequently has two outputs: one sending payment and one returning change. Thirdly, the payer and payee(s) of a transaction are identified through public-keys from public private key pairs. However, a user can have multiple public-keys.

The above three features are public availability of Bitcoin transactions [2], the input-output relationship between transactions and the reuse and couse of public-keys, provide a basis for two distinct network structures: the TX network and the users' network. The Tx network represents the flow of Bitcoins between transactions over time. Each vertex represents a transaction and each directed edge between a source and a target represents an output of the transaction corresponding to the source that is an input to the transaction corresponding to the target. Each directed edge also includes a value in Bitcoins and a timestamp (Figure 1). The users' network represents the flow of Bitcoins between users over time. Each vertex represents a user and each directed edge between a source and a target represents an input-output pair of a single transaction where the input's public-key belongs to the user corresponding to the source and the output's public-key belongs to the user corresponding to the target. Each directed edge also includes a value in Bitcoins and a timestamp (Figure 2 & 3).

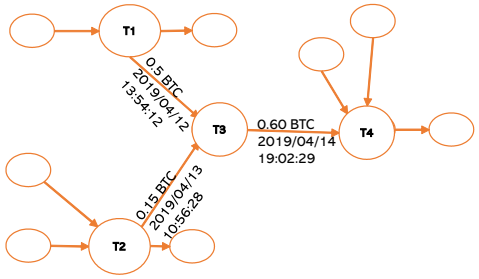


Figure 1. An example of TX network. Each cycle vertex represents a transaction and each directed edge represents a flow of BTC from an output of one transaction to an input of another.

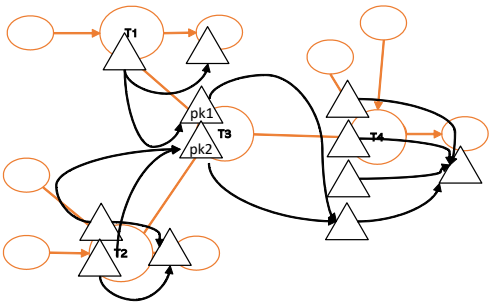


Figure 2. An example of Bitcoin users' network. Each triangle vertex represents a public-key and each directed edge between the triangle vertices represent a flow of Bitcoins from one public-key to another.

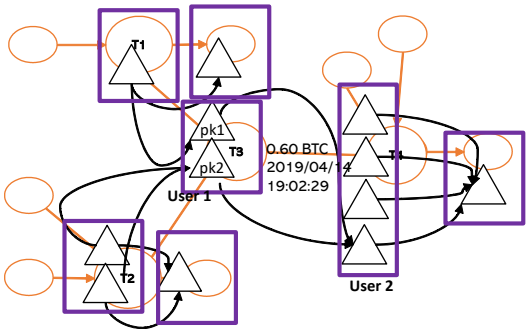


Figure 3. An example Bitcoin users' network. Each rectangular vertex represents a user and each directed edge between the rectangular vertices represents a flow of Bitcoins from one user to another.

III. De-anonymizing Bitcoin

In this section, we will be describing the used methodology for de-anonymizing Bitcoin. Our proposed approach of de-anonymizing attack has four main phases.

1. Collect the historical Bitcoin transactions databased, and dynamically collect a list of all bitcoin nodes S.
2. Generate the de-anonymizing attack for a specific set of N nodes.
3. Mapping users (public-keys) to nodes
4. Mapping transactions to nodes

All the for attack phases will be described in details in the upcoming sub-sections.

2.1 Collection of Data

We based our Bitcoin mapping system on mainly using the data provided by the Bitcoin network [4], we did use the Bitcoin client and our created customized client to download all the transactions history and gather all Bitcoin active nodes' data and peers' data. In this phase of the attack. Our attacker node first collects the entire history of transactions. Then, collect the list of peers by querying all known peers with a GETADDR message. Each address IP in the response ADDR message can be checked if it is reachable by establishing a TCP connection and sending a VERSION message. If it is reachable, the IP is designated as a server. The attacker node initiates the procedure by querying a small set of nodes and continue by querying the newly received IP addresses. The adversary establishes connections number Cn to each server (we suggest 100 for the size of the current Bitcoin network).

For each discovered peer, we will create a list $P = (IP; Id; PK)$, where IP is the IP address of a peer or its ISP, Id distinguishes clients sharing the same IP, and PK is the pseudonym used in a transaction (hash of a public key).

2.2 De- anonymization attack

In the second phase, the attacker node selects an N number of nodes to reveal their identities. Then, The attacker node collect of each given node in the N set the "peer.dat" list and query the Txs' history. we should emphasize here, the attacker node should keep the connection alive with all the bitcoin nodes, so whenever a node propagates a new transaction to the network we will be hearing it form the source and not through the gossiping flooding.

2.3 Mapping Users to nodes locations

In this phase of the attack, the attacker node identifies the entry nodes of the users that are connecting to the network. Supplied with the list S of addresses, the attacker node runs the topology discovering mechanism described in the following Section 4. To uniquely identify a user we need to estimate the number of the entry nodes needed by each user. eP denote the set of entry nodes for P (a discovered peer). In here we need to emphasize that $eP1 \neq eP2$ even if $P1$ and $P2$ share the same IP address. For each P advertising its address in the network, the attacker node obtains a set of $eP'_p \subseteq eP_p$. Since there are about 8.10^3 possible entry nodes out of 10^5 total peers (servers and clients together), the collisions in eP'_p are unlikely if every tuple has at least 3 entry nodes (1):

$$\frac{10^5 .10^5}{(8 .10^3)^3} \ll 1 \quad (1)$$

Therefore, 3 entry nodes uniquely identify a user, the attacker nodes will proceed the same for a large percent of users. The attacker node will add the eP to its database and proceeds the final phase of the attack (Step 4).

2.4 Mapping Tx to nodes locations

This step runs in parallel to previous phases. Now the attacker node tries to map the transactions appearing in the network with sets eP'_p obtained in second phase. The attacker node listens for INV messages with transaction hashes received overall its connections and for each transaction Tx it collects rTx the first P addresses of Bitcoin nodes that forwarded the INV message. The attacker node then compares eP with rTx , and the matching Tx to nodes giving the pairs (eP; rTx). By using this technique there could be many variants for the matching procedure, such as the attacker node organizes all possible 3-tuples from all sets *ePi* and looks for their appearances in rTx . If there is a match, it gets a pair (R; Tx). If there is no match, the attacker node considers 2-tuples and then 1-tuples. Several pairs $(eP_i; Tx)$ can be suggested at this stage, but we can filter them with later transactions.

IV. Topology Discovery

In the case where a Bitcoin user is behind a TOR, or he/she banning the incoming connections (only allowing an outgoing connection to 8 peers). The method proposed to learn the data of these nodes is based on the fact that whenever a client Ci establish an outgoing connection its address aCi is then sent to the entry peer. If the attacker node is already connected to entry peers, the attacker node will be able to know the aCi (depends on the attacker's connections number). To achieve this, the attacker node needs to connect to S Bitcoin nodes. Where S is close to the total number of nodes. in addition, for each learned aCi, the log set N′ of nodes that forwarded the aCi to the attacker node, will specify it as the subset entry peer N'_{aCi}

Here are some details. When the attacker announces the aCi, each Bitcoin node chooses a set of responsible peers to forward the address. The attacker then will establish a number of connections to each node in the network hoping that its nodes will replace some of the responsible nodes for address aCi. When client *C* connects to one of its entry nodes *eP1*, it advertises its address. If one of the attacker's nodes replaced one of the responsible nodes, then the attacker will learn that client *C* might be connected to node *eP1*. If the responsible nodes did not change address, the aCi won't be propagated further in the network.

The used method has the following two limitations. First, the entry nodes might forward the client's address to a non-attacker peer. Second, a client cannot connect to all his entry nodes simultaneously, but there is a time gap between connections. In both cases, the advertised address reaches the attacker's machines via peers that are not entry nodes, which yields false entries in N'_{aCi}.

IV. Clustering Algorithm

Now, having describe the mapping techniques, we need to explain the used clustering approach. In searching for an appropriate clustering algorithm that will be able to map all Tx and public key within the Bitcoin network to the nodes locations, we had to take into consideration the following:

Noisy data: the collected data would be noisy, such as, it will contain nodes that wouldn't belong to any cluster because we expected large numbers of users with only one address. Therefore, we needed a clustering algorithm that could harmonize noisy data.

Varying of clusters numbers: Because of the absence of any ground-truth data on clusters of addresses, we were unable to definitively establish a well-defined number of clusters.

Varying of clustering density: Since some users clusters would be connected by a variety of similarity measures while others would only be joined with a heuristic, these clusters would also likely vary in density.

Floating and large data: Because the collected data is non-positional and every huge (Total Tx combined with the total public keys and nodes data), we need an algorithm that could accurately and quickly partition the massive fed dataset into clusters.

Therefore, after evaluating several deep-learning based algorithms, we decided to create a modified unsupervised clustering algorithm that combine K-Means and DBSCAN algorithms. This algorithm satisfies all our four considerations, upon K-Mean [5] it can register points as noise, it decides on the number of clusters present in the graph automatically, it improves upon DBSCAN [6] by being able to adequately deal with varying-density clusters, it doesn't require positional information, and it performs adequately quickly. The algorithm works by first modifying the graph, pushing outliers further away to highlight particularly dense clusters. Then, it organizes these nodes into a cluster hierarchy of connected components based on distance and performs single-linkage clustering on the remaining points. It decides when to stop links from clustering further by building a condensed version of the cluster tree and identifying the most reasonable cutoff points for each branch.

V. Experimental Results

In order to evaluate the proposed users de-anonymizing method, we gathered the entire history of Bitcoin transactions from the first transaction on the 3rd January 2009 up to and including the last transaction that occurred on the 09th April 2019. We gathered the dataset using the Bitcoin Core client [3]. The dataset comprises of 400,042,017 transactions between over 32 million public-keys. However, for nodes data collection we did uses a modified python version of Bitcoin core that allows us to gather all the peers.dat from all active bitcoin nodes (we had detected over 162,000 IP addresses).

5.1 Bitcoin Tx and pseudonyms mapping:

For the preliminary evaluation, we computed the mapping techniques for the first 200K Bitcoin addresses. We then inspected the 100 addresses with the highest weight, and attempt to label them using our local data based. During the preliminary evaluation, we found that 76% of those highly weighted transactions belongs to betting game services, such as SatoshiDICE, MegaDice. While the remaining 14% of those highly weighted transactions belongs to donation services.

5.2 Bitcoin nodes mapping:

The nodes can be easily grouped by mean of geographic regions (Geolocation) or Autonomous Systems (AS). We used an IP Geolocation API [15] to obtain both information regarding the geographical location of the node in Bitcoin main-net and their distribution among autonomous systems. Table 1 provides data concerning the Bitcoin nodes distribution over the top 10 countries. During the last scan of the network, most of the discovered node IP addresses were located in the United States with more than 28%, China 24.7%. The Tor IP addresses and some other nodes IPs around 2.7% of the discovered IP addresses during the last scan couldn't be geolocated due to their used internet infrastructure. Figure 4 provides the distribution of Bitcoin active reachable nodes among the 1,639 discovered autonomous systems. As shown in the graph more than 60% of all the discovered reachable nodes in Bitcoin reside in only 50 ASs, we also noticed that 7% of the AS contained a SINGLE Bitcoin node.

TABLE I. DISTRIBUTION OF BITCOIN NODE IP ADDRESSES BY COUNTRY

Country	Total Nodes	Percentage
UNITED STATES	5353	28.30%
PEOPLE'S REPUBLIC OF CHINA	4663	24.70%
GERMANY	2185	11.60%
FRANCE	814	4.30%
NETHERLANDS	765	4.10%
UNITED KINGDOM	531	2.80%
RUSSIAN FEDERATION	512	2.70%
CANADA	492	2.60%
ITALY	220	1.20%
SWITZERLAND	211	1.10%

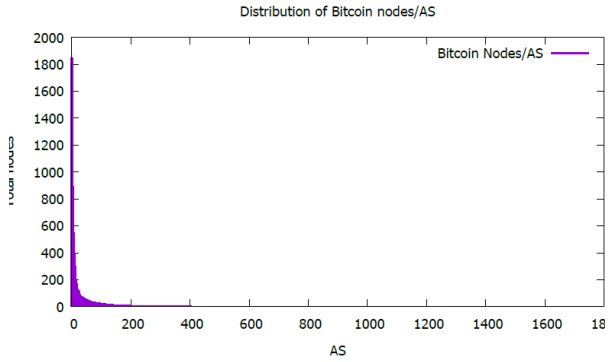


Figure 4. Distribution of Bitcoin nodes by AS

VI. Discussion

Bitcoin data is continuous, huge, and changing frequently. Thus, the process of mapping Bitcoin TX/public-keys to nodes location is quite challenging, due to the protocol implementation, data nature and internet infrastructure. In this section, we will underline some of the strengths and weaknesses of the proposed solution.

To make the proposed attack de-anonymization less detectable and prevent the used attack node from being denied, the attacker node needs to keep a significant number of connections to Bitcoin nodes without sending a large amount of data, so that all connection look like they came from different unrelated clients and the same set of IP addresses can be used for different attacker nodes.

The technique proposed in the paper provides unique identification of Bitcoin clients for many sessions. Therefore, if a client makes multiple transactions during one session they can be linked together with very high probability. Note that this is done even if the client uses totally unrelated public keys/Bitcoin wallets, which would be totally unachievable through transaction graph analysis [8]. Moreover, by applying the multi-linkage stages, we can easily distinguish all the different clients even if they come from the same ISPs, hidden behind the same NAT or firewall address.

VII. Related Work

Past work into investigating anonymity in the Bitcoin blockchain has generally proceeded along one of two paths:

Identifying Users: Most implemented clustering systems find heuristics that roughly correspond to co-ownership of addresses in the blockchain, By either using public information, Meiklejohn et al. attempt to identify users manually by transacting with companies and parsing forums [9]. Kichiji and Nishibe [10] used the collected certificates to derive a network structure that represented the ow of currency during the period. They showed that the cumulative degree distribution of the network obeyed a power-law distribution, the network had small-world properties (the average clustering coefficient was high whereas the average path length was low), the directionality and the value of transactions were significant features, and the double-triangle system [10] was effective.

Network Analyses: Biryukov et al [11] investigated whether attackers could discover user information by exploiting loopholes in the bitcoin network. Narayanan and Shmatikov [12] and Backstrom et al. [13] consider privacy attacks which identify users using the structure of networks and show the difficulty in guaranteeing anonymity in the presence of network data. Crandall et al. [14] infer social ties between users where none are explicitly stated by looking at patterns of `co-incidences' or common network co-occurrences.

Our approach involves unsupervised algorithms that map Bitcoin users pseudonyms to read identities based on nodes clustering, We first map public keys to nodes

and then map transactions to nodes, the clustering is based on the Txs' history and nodes data provided by the network, In doing so, we extend upon past work done in the space in a few salient ways: We contribute a new 'loose heuristic' for address similarity. We represent information for both TX/pseudonyms mapping to bitcoin node, and We provide the geolocation for the mapped nodes. We use the unsupervised clustering algorithms that vary in density and do not require approximations of cluster numbers.

VIII. Conclusion and future work

In this paper, we proposed a novel approach to map Bitcoin to clients node; for this mean, we combined several mapping techniques. First, we performed a de-anonymization attack, which allows our local nodes to collect all the Bitcoin historical transaction database, discover and establish a connection with every active node in the network. Then, by using special mechanisms, we map users (public-keys) to nodes and map transactions to nodes and finally geolocate all the discovered nodes. Our approach is based on an unsupervised clustering algorithm that combines both K-Means and DBSCAN algorithms, upon K-Mean it can identify the noise in data, and predicted the needed number of clusters present in the graph automatically, as well known, the DBSCAN algorithm doesn't require positional information, therefore, we used it to deal with varying-density clusters.
The presented approach is still under development. In the future, we aim to expand our search area to include TOR network and coin-mixing protocols.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2015R1D1A1A01059786), and Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.2018-0-00539, Development of Blockchain Transaction Monitoring and Analysis Technology).

References

[1] Bitcoin: A Peer-to-Peer Electronic Cash System. Available Online, https://bitcoin.org/bitcoin.pdf , Accessed 04/13/2019.

[2] Bitcoin Transactions, Available Online, https://en.bitcoin.it/wiki/Transaction , Accessed 04/13/2019.

[3] Bitcoin Core, Available Online, https://bitcoin.org/en/download , Accessed 04/13/2019.

[4] Bitcoin Protocol, Available Online, https://en.bitcoin.it/wiki/Protocol_documentation , Accessed 04/13/2019.

[5] k-Means Clustering, Available Online, https://jakevdp.github.io/PythonDataScienceHandbook/05.11-k-means.html , Accessed 04/13/2019.

[6] DBSCAN: density-based clustering for discovering clusters in large datasets with noise - Unsupervised Machine Learning, Available Online, http://www.sthda.com/english/wiki/wiki.php?id_contents=7940 , Accessed 04/13/2019.

[7] Decker et al. Information Propagation in the Bitcoin Network, 13-th IEEE International Conference on Peer-to-Peer Computing, Sept. 2013.

[8] Michael Fleder et al. Bitcoin Transaction Graph Analysis, Information Retrieval (cs.IR); Social and Information Networks (cs.SI), Feb 2015.

[9] Meiklejohn, Sarah et al. "A Fistful of Bitcoins." Proceedings of the 2013 conference on Internet measurement conference - IMC '13, 2013, doi:10.1145/2504730.2504747.

[10] N. Kichiji and M. Nishibe. Network Analyses of the Circulation Flow of Community Currency. Evolutionary and Institutional Economics Review, 4(2):267{300, 2008.

[11] Biryukov, Alex et al. "Deanonymisation of Clients in Bitcoin P2P Network." Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14, 2014.

[12] A. Narayanan and V. Shmatikov. De-anonymizing Social Networks. In Proceedings of the 30th Symposium on Security and Privacy, pages 173{187. IEEE, 2009.

[13] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore Art Thou r3579x?: Anonymized Social Networks, Hidden Patterns, and Structural Steganography. In Proceedings of the 16th International Conference on World Wide Web, pages 181{190. ACM, 2007.

[14] D. Crandall, L. Backstrom, D. Cosley, S. Suri, D. Huttenlocher, and J. Kleinberg. Infer-ring Social Ties from Geographic Coincidences. Proceedings of the National Academy of Sciences, 107(52):22436, 2010.

[15] IP Geolocation API, [Online], Available: https://ipstack.com/

[16] Chainalysis: Darknet Market Activity Nearly Doubled Throughout 2018, [Online], Available: https://bitcoinmagazine.com/articles/chainalysis-darknet-market-activity-nearly-doubled-throughout-2018/

[17] Jason Luu et al. The Challenge of Bitcoin Pseudo-Anonymity to Computer Forensics, [Online], Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2671921

[18] S.M. Avdoshin et al. Bitcoin Users Deanonymization Methods, [Online], Available: https://publications.hse.ru/mirror/pubs/share//direct/217565576

[19] Husam Al Jawaheri et al. When A Small Leak Sinks A Great Ship: Deanonymizing Tor Hidden Service Users Through Bitcoin Transactions Analysis. [Online], Available: https://arxiv.org/pdf/1801.07501.pdf

비트코인 네트워크의 불법거래 탐지 연구

이채현 O, Sajan Maharjan, 고경찬, 홍원기

포항공과대학교 컴퓨터공학과

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

Research of Detecting Illegal Transactions on Bitcoin Network

Chaehyeon Lee O, Sajan Maharjan, Kyunchan Ko, James Won-Ki Hong

Department of Computer Science and Engineering, POSTECH

요 약

비트코인은 블록체인 기술을 기반으로 한 암호화폐 기술로 제 3 자의 개입 없이 모든 네트워크 참여자가 동일한 데이터를 포함하는 분산 원장을 가진다. 데이터 무결성, 데이터 불변성, 주소 익명성을 통한 프라이버시 보장의 장점을 가지는 비트코인은 가상화폐로써 국경을 뛰어넘는 지불 수단으로 활용되고 있다. 하지만 주소와 연관된 네트워크 참여자의 실제 신원을 알 수 없는 익명성 때문에 암거래 시장의 불법 거래에 비트코인이 악용되고 있다. 이에 본 연구에서는 비트코인 네트워크에서 발생하는 불법 거래를 탐지하기 위한 기계학습 기반의 방법론을 소개한다.

I. 서 론

비트코인 [1] 은 2008 년 사토시 나카모토에 의해 처음으로 세상에 등장했다. 비트코인은 분산화된 P2P 네트워크 형태를 가지며, 네트워크의 참여자가 동일한 공개거래장부를 유지한다. 컴퓨팅 자원을 이용해 특정 범위 내의 해시 결과 값을 찾는 '채굴' 과정을 통해 비트코인을 생성할 수 있으며, 키를 소유한 사용자들은 거래 검증 과정을 거쳐 누구나 비트코인을 소비할 수 있다. 또한 공개거래장부를 통한 데이터 무결성, 한 번 기록된 데이터를 추후에 수정/삭제할 수 없는 데이터 불변성, 주소를 이용해 실 소유주를 유추할 수 없는 주소 익명성을 보장한다. 암호학과 분산 시스템을 기반으로 한 1 세대 블록체인 [2] 인 비트코인은 가상화폐라는 새로운 화폐의 패러다임을 제시함과 동시에 새로운 지불 수단으로 사용되고 있다. 기존 화폐처럼 물건을 구매하거나, 다른 사람 또는 기관에 송금할 수 있으며, 국경에 국한되지 않는 글로벌한 통화수단으로 사용될 수 있다.

하지만 비트코인 주소와 연관된 네트워크 참여자의 실제 신원을 확인할 수 없는 익명성으로 인해, 비트코인이 불법적인 거래에 악용되고 있다. 암호화폐가 전자 지불 수단으로 사용되는 사례가 증가함에 따라 마약 및 무기 거래, 돈 세탁, 스캐밍(Scamming)과 같은 범죄 활동의 시도가 빈번하게 나타나고 있다. 실제로, 블랙마켓의 아마존과 같은 Silk Road [3]가 블록체인의 익명성을 기반으로 운영되고 있으며, 웹사이트를 통한 마약 및 무기 불법 거래금액이 연간 120 만 달러에 달한다고 한다. [4] 현재 비트코인은 이러한 불법거래에 가장 많이 사용되는 암호화폐 플랫폼이며, 빈번하게 발생하는 불법거래는 비트코인 관련 법률 제정을 저해하는 요인으로 작용하고 있다. 따라서 비트코인 네트워크 상에서 발생하는 불법적인 거래를 사전에 차단할 수 있는 방안이 필요하고, 이에 본 논문에서는 기계학습을 기반으로 하여 불법거래를 탐지할 수 있는 방법론을 소개한다.

II. 관련 연구

Deepak [5]은 비트코인 절도(thefts)와 관련된 사용자의 특성을 파악하고, 유사한 행동을 하는 사용자를 식별하기 위한 기계학습 기반의 시스템을 제안했다. 절도, 사기성 활동을 감지하기 위해 연관된 비트코인 트랜잭션 정보를 추출하고, 추출된 특징들을 분류하기 위해 k-means 기반의 Unsupervised 클러스터링 모델을 활용하였다.

Kentaroh [6]는 HYIP (High Yielding Investment Program)와 관련된 비트코인 주소를 수집하기 위해 트랜잭션 패턴을 분석하였다. 비트코인 주소와 연관된 트랜잭션의 수, 채굴된 블록의 수 등의 특징을 추출하고, 비트코인 주소를 HYIP 또는 non-HYIP 로 라벨링 하여 supervised 학습을 통해 사이버범죄 집단을 분류하였다.

비트코인 범죄와 연관된 비트코인 주소 및 군집을 식별, 분류하는 연구는 활발히 진행되었으나, 트랜잭션 classification 에 중점을 둔 사전 연구를 찾을 수 없었다. 본 연구에서는 범죄와 연관된 비트코인 트랜잭션이 가지는 공통된 특징을 파악하여, 불

법거래를 탐지하고자 한다.

III. 불법거래 탐지 시스템

비트코인 네트워크에서 발생하는 불법적인 거래를 탐지하기 위해 4 단계에 걸친 방법론을 소개한다. 1. 트랜잭션 수집, 2. 특징 추출, 3. 기계학습 모델 구현 및 학습, 4. 검증 및 결과 도출 단계가 이에 해당하며 그림 1 과 같은 절차를 따른다.

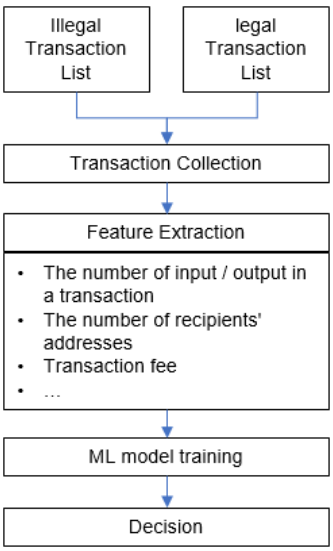


그림 1 불법거래 탐지 절차

1. 트랜잭션 수집

기계학습 모델을 구현하기에 앞서, 불법적인 트랜잭션과 합법적인 트랜잭션의 리스트를 수집한다. 일부 웹사이트에서는 앞서 언급한 Silkroad 에서 발생시킨 거래의 hash 및 악의적인 목적으로 비트코인을 이용한 비트코인 주소를 일부 제공하고 있다. 또한 포럼 사이트에는 교환소, 메모리풀, 겜블링 등 특정 집단에서 사용된 데이터를 카테고리 별로 공개하고 있다. 따라서 웹 크롤링을 이용해 연관된 불법 트랜잭션과 합법 트랜잭션의 리스트를 얻을 수 있다.

	A
1	ae018d96[illegible]3f0f4568f5fe5415e315b430ed5fd5b8b1150479b4219c005145f52
2	26641f6eb050a256791801fd9b59e40b907a891ee044a0d1713af767b85b25cd
3	66d9e2d03186e7184c26b7d4522f7b160c52630941dca14bfff423a4c1b5d8e9
4	45059044bc476466e8a5c1ee51280bcad8f6fd42fd7db367cd23a56c36468c3a
5	cfc2e5716edf644bd9cf633a856c7630341ce54c7bb138334ff899d965983e08
6	c105f56b0931307c2eddbe285c4eaac06b7200c799543b694c1a1be0ec27e7cf
7	804623f3e55703fdffeddfbf80e4830ef0f066b4e1488d32c39aed3555d80da7
8	d8d6927e0a9007d97985ed728cfb4530d11117712330c73c849ce3eac8800cc3
9	53d3021c2d7c0745e08478a5a78c432e372d017a203fe7cc3d9c00cac8fc9872
10	6e97d9c241bb3e1c96a16f267fee8322f5186269d099ad7545077cb65d23b5f1
11	efd57f85ac38f489cdef4d3ac6f9e20c9395f1af3ca6d8eaf478335e25c4456c
12	915e3a2a26f75739c8df2b9d13016f497a73e0aa9dbbf454b49bad3838b94521
13	6b774ad1cf26d938f23dd094716dfa745eef6ad7372cf2933854639bbbc8276a
14	cd4c7a30b15c3b8f738bd3155f309a0eb9e053972b510bb63664a9ea89806070
15	81b3593f784f4ccfbc9fbae5c00cc06a09daf0f631854a579112c7ed51468890
16	730e2d30f38a48602599281686a4dcd3918047e87540ddbb880d524b7382d1d9

그림 2 수집된 불법 트랜잭션 리스트

2. 특징 추출

트랜잭션 리스트를 수집하고 나면 불법적인 거래가 갖는 특성을 파악하기 위해 주요 특징들을 추출한다. 불법적인 트랜잭션들은 특정 주소로 비트코인을 전송하거나, 블록에 빨리 포함되기 위해 높은 거래 수수료를 지불하는 등의 공통적인 특징을 가질 수 있다. 따라서 구체적인 트랜잭션 정보로부터 머신러닝 모델을 학습시키기 위한 주요 특징들을 추출한다.

먼저, 첫 번째 단계에서 얻은 트랜잭션 ID 값을 키워드로 하여 JSON-RPC 를 이용해 구체적인 트랜잭션 정보를 얻어온다. 이를 바탕으로, 트랜잭션에 포함된 input 과 output 의 개수, 거래의 결과로 비트코인을 수신하게 될 비트코인 주소의 수, 거래 수수료, 거래의 크기, 거래량 등을 주요 특징으로 추출한다 (그림 3).

합법적인 트랜잭션과 불법적인 트랜잭션은 non-Illegal (Legal) 또는 Illegal 로 라벨링 되어 추후 supervised learning 모델을 통해 학습된다. 추출된 특징들은 기계학습 모델을 통해 학습시키기 편리한 형태인 CSV 형태로 저장되며 추후 제대로 학습되었는지 검증하는 과정에서 활용될 수 있도록 수집된 트랜잭션들을 training set 과 test set 으로 나눈다.

	A	B	C	D	E	F	G	H	I	J	K
1	txid	vin	vout	vin_value	vout_value	fee	in_addr	out_addr	size	value	legal
2	ae018d96[illegible]	1	2	0.007559	0.007522	3.66E-05	1	2	373	0.007522	0
3	26641f6eb[illegible]	1	2	0.007727	0.007691	3.66E-05	1	2	373	0.007691	0
4	66d9e2d0[illegible]	1	2	0.007896	0.007859	3.66E-05	1	2	373	0.007859	0
5	45059044[illegible]	1	2	0.00847	0.008433	3.65E-05	1	2	369	0.008433	0
6	cfc2e5716[illegible]	1	2	0.008664	0.008628	3.65E-05	1	2	372	0.008628	0
7	c105f56b0[illegible]	2	2	0.010104	0.010036	6.74E-05	1	2	670	0.010036	0
8	804623f3e[illegible]	1	2	0.00063	0.000594	3.64E-05	1	2	369	0.000594	0
9	d8d6927e[illegible]	1	2	0.012415	0.012396	1.89E-05	1	2	372	0.012396	0
10	53d3021c2[illegible]	1	2	0.089659	0.089587	0.000072	1	2	372	0.089587	0
11	6e97d9c24[illegible]	1	17	1.006517	1.006334	0.000183	1	17	729	1.006334	0
12	efd57f85a[illegible]	2	151	20.5324	20.53106	0.001344	2	151	5372	20.53106	0
13	915e3a2a2[illegible]	1	750	0.00013	3E-05	0.0001	1	750	25691	3E-05	0
14	6b774ad1[illegible]	1	750	0.00013	3E-05	0.0001	1	750	25691	3E-05	0
15	cd4c7a30[illegible]	1	750	0.00013	3E-05	0.0001	1	750	25691	3E-05	0
16	81b3593f7[illegible]	1	750	0.00013	3E-05	0.0001	1	750	25691	3E-05	0
17	730e2d30f[illegible]	1	750	0.00013	3E-05	0.0001	1	750	25691	3E-05	0
18	79b80066[illegible]	1	750	0.00013	3E-05	0.0001	1	750	25691	3E-05	0
19	b6433b02[illegible]	1	749	0.000107	7.49E-06	0.0001	1	749	25658	7.49E-06	0
20	1c80ad7c0[illegible]	1	749	0.000107	7.49E-06	0.0001	1	749	25657	7.49E-06	0
21	ab052c7a3[illegible]	1	749	0.000107	7.49E-06	0.0001	1	749	25657	7.49E-06	0
22	5f4feab6ft[illegible]	1	749	0.000107	7.49E-06	0.0001	1	749	25657	7.49E-06	0
23	bcd5c9eff[illegible]	1	749	0.000107	7.49E-06	0.0001	1	749	25657	7.49E-06	0
24	b29ffb7e3[illegible]	1	749	0.000107	7.49E-06	0.0001	1	749	25657	7.49E-06	0
25	4debeeb5[illegible]	1	750	0.00013	3E-05	0.0001	1	750	25691	3E-05	0

그림 3 불법 트랜잭션 특징 추출

3. 머신러닝 모델 구현 및 학습

라벨링 된 트랜잭션들의 추출된 특징을 학습시키고, 주어진 특정 트랜잭션의 불법/합법 여부를 판단하기 위해 supervised 기반의 분류 모델을 구현한다. 분류 모델 알고리즘으로 Random forest classification [7] 또는 사람의 신경망을 기반으로 한 인공 신경망 (Neural network) 등을 활용할 수 있다. 수집된 데이터 셋을 이용해 구현한 분류 모델을 학습시킨다. 이때 합법 데이터와 불법 데이터의 분포 비율을 다르게 설정하여 여러 번 실험을 진행한다. 추출된 특징들 중 결과에 영향을 많이 미치는 특징을 찾고, 해당 특징에 더 높은 가중치를 적용하는 방식으로 실험을 진행하여 모델의 정확도를 높일 수 있다.

4. 검증

학습이 완료되고 나면, test set 을 이용해 기계학

습 모델이 제대로 구현되었는지 검증한다. 구현이 잘 되었다면, 해당 모델을 활용해 주어진 특정 트랜잭션이 불법적인지 합법적인지 분류할 수 있다.

IV. 결론 및 향후 연구

비트코인 네트워크의 불법거래 탐지 기술은 비트코인을 불법적인 활동에 악용하려는 사용자의 시도를 사전에 인지하고, 차단하기 위해 필수적이다. 이에 본 논문에서는 기계학습을 이용해 불법거래를 탐지할 수 있는 방법론에 대해 소개했다. 향후 연구에서는 제안한 시스템을 구현하고, 성능 검증을 통해 시스템의 정확도를 측정 및 개선할 예정이다. 또한 블록체인 시스템을 이더리움으로 확장하여 이더리움 네트워크에서 발생한 해킹 사례들을 분석하고, 이를 기반으로 이더리움의 불법거래를 탐지하기 위한 연구를 진행할 계획이다.

ACKNOWLEDGMENT

이 논문은 2019 년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구 임 (No.2018-0-00539)

참 고 문 헌

[1] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008).
[2] Swan, Melanie. Blockchain: Blueprint for a new economy. "O'Reilly Media, Inc."(2015.)
[3] Harvey, Campbell R. "Bitcoin myths and facts." (2014).
[4] https://www.gwern.net/Silk-Road
[5] Zambre, Deepak, and Ajey Shah. "Analysis of Bitcoin network dataset for fraud." *Unpublished Report* (2013).
[6] K. Toyoda, T. Ohtsuki and P. T. Mathiopoulos, "Identification of High Yielding Investment Programs in Bitcoin via Transactions Pattern Analysis," *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Singapore, 2017, pp. 1-6.
[7] M. Pal. (2005) Random forest classifier for remote sensing classification. *International Journal of Remote Sensing* 26:1, pages 217-222.

주성분 분석을 적용한 클러스터링을 이용한 비트코인 네트워크 분석 방법

신무곤, 백의준, 구영훈, 지세현, *박준상 김명섭
고려대학교, *LG Electronics
{tm0309, pb1069, gyh0808, sxzer, tmskim}@korea.ac.kr, *junsang.park@lge.com

Bitcoin Network Analysis Method Using Clustering with Principal Component Analysis

Mu-Gon Shin, Ui-Jun Baek, Young-Hoon Goo, Se-Hyun Ji, *Jun-Sang Park, Myung-Sup Kim
Korea Univ., *LG Electronics

요 약

블록체인 기술을 기반으로 만들어진 온라인 암호화폐 비트코인은 개인, 기업, 정부 등 모두의 관심을 끌고 있다. 지난 몇 년간 블록체인 기술과 암호화폐에 대한 관심이 꾸준히 증가함에 따라 암호화폐의 거래량 및 시장규모는 놀라운 속도로 증가했다. 이에 따라 블록체인 네트워크와 블록, 트랜잭션에 대한 분석 및 모니터링 방안은 중요한 이슈가 되고 있다. 본 논문에서는 비트코인 네트워크 분석 방안으로 차원축소를 적용한 클러스터링 방법을 제안한다. 제안된 방법은 본 연구팀이 수집한 비트코인 내의 블록 데이터에 PCA를 적용한 K-means알고리즘을 이용한 분석 방법을 적용한다.

Ⅰ. 서 론

2008년 10월 사토시 나카모토가 개발한 비트코인(bitcoin)은 블록체인 기술을 기반으로 만들어진 온라인 암호화폐이다[1]. 지난 몇 년간 블록체인 기술과 암호화폐에 대한 관심이 꾸준히 증가함에 따라 암호화폐의 거래량 및 시장규모는 놀라운 속도로 증가했다. 2019년 4월 기준, 비트코인의 하루 평균 거래량(트랜잭션 수)는 약 38만건에 달한다. 비트코인의 거래량이 증가하고 블록체인에 대한 관심이 깊어지고 있지만, 블록체인에 대한 모니터링 및 분석에 대한 연구는 많지 않다. 비트코인을 포함한 암호화폐를 통한 불법적인 거래 등이 늘어남에 따라 암호화폐 블록과 트랜잭션에 대한 모니터링하고 분석하는 것은 매우 중요하다.

클러스터링 알고리즘 중 하나인 K-Means 알고리즘은 주어진 데이터를 k개의 클러스터로 묶는 알고리즘으로, 각 클러스터와 거리 차이의 분산을 최소화하는 방식으로 동작한다[2]. 여러 개의 feature들로 구성된 비트코인의 블록 및 트랜잭션 데이터는 클러스터링을 통하여 여러 군집으로 묶일 수 있다. 또한 효과적인 클러스터링을 위하여 feature를 선택하는 것은 매우 중요하다. 그리고 클러스터링 된 데이터를 시각화 하는데 있어 차원축소도 중요한 이슈가 될 수 있다.

차원축소 알고리즘 중 하나인 PCA(Principal Component Analysis)는 고차원의 데이터를 저차원의 데이터로 환원시키는 기법이다[3]. 블록 데이터들이 여러 개의 고차원 데이터로 표현 되기 때문에 PCA를 활용하여 고차원의 블록 데이터들을 저차원의 데이터로 변환하여 이 정보들을 클러스터링에 이용하였다. 또한 저차원으로 변환된 데이터들은 2차원이나 3차원의 그래프로 시각화하여 데이터들의 분포를 명확히 알 수 있다.

본 논문은 서론에서 연구 배경과 목표를 서술하고, 본론에서 비트코인 블록데이터 분석을 위한 PCA를 적용한 K-Means 알고리즘을 제안한다. 제안하는 방법은 실험을 통해 결과를 검증한다.

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

Ⅱ. 본론

본 장에서는 PCA와 K-Means에 대해 소개하고 본 연구팀이 수집한 비트코인 블록 데이터와 PCA를 적용한 K-Means 클러스터링 방법에 대해 언급한다.

2.1 PCA(Principal Component Analysis)

본 절에서는 PCA에 대해 언급한다. PCA(Principal Component Analysis)는 데이터의 분산(variance)을 최대한 보존하면서 서로 직교하는 새 기저(축)를 찾아, 고차원 공간의 표본들을 선형 연관성이 없는 저차원 공간으로 변환하는 기법이다.

$$\arg\min_{\hat{x}} ||x - U\hat{x}||^2 \quad \text{[식 1]}$$

식 1에서 U는 역변환 행렬을 의미하고 χ는 원래의 벡터를 의미한다. 또한 $\hat{\chi}$은 원래 벡터와 가장 비슷해지는 차원축소 벡터를 의미한다. 이러한 과정을 통하여 PCA는 원래 입력 벡터와 가장 비슷한 축소된 새로운 차원의 벡터를 구한다.

PCA는 다음과 같은 단계로 이루어진다.

1. 학습 데이터셋에서 분산이 최대인 축(axis)을 찾는다.
2. 이렇게 찾은 첫번째 축과 직교(orthogonal)하면서 분산이 최대인 두 번째 축을 찾는다.
3. 첫 번째 축과 두 번째 축에 직교하고 분산을 최대한 보존하는 세 번째 축을 찾는다.
4. 1~3과 같은 방법으로 데이터셋의 차원(특성 수)만큼의 축을 찾는다.

2.2 K-Means Clustering

본 절에서는 K-Means에 대해 언급한다. 주어진 데이터를 k개의 클러스터로 묶는 알고리즘으로, 각 클러스터와 거리 차이의 분산을 최소화하는 방식으로 동작한다. 이 알고리즘은 자율 학습의 일종으로, 레이블이 달려 있지 않은 입력 데이터에 레이블을 달아주는 역할을 수행한다.

n개의 d-차원 데이터 오브젝트 (x1, x2, ⋯, xn) 집합이 주어졌을 때, K-Means 알고리즘은 n개의 데이터 오브젝트들을 각 집합 내 오브젝트 간 응집도를 최대로 하는 k개의 집합 S = {S1, S2, ⋯, Sk} 으로 분할한다. 다시 말해, μi가 집합 Si의 중심점이라 할 때

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 \quad \text{[식 2]}$$

각 집합별 중심점 ~ 집합 내 오브젝트간 거리의 제곱합을 최소로하는 집합 S를 찾는 것이 목표이다.

K-Means는 다음의 두 단계를 반복한다.

1. 클러스터 설정: 각 데이터로부터 각 클러스터들의 중심까지의 유클리드 거리를 계산하여, 해당 데이터에서 가장 가까운 클러스터를 찾아 데이터를 배당한다.

2. 클러스터 중심 재조정: 클러스터 중심을 각 클러스터에 있는 데이터들의 무게중심 값으로 재설정해준다.

K-means 알고리즘은 클러스터 개수 k값을 파라미터로 지정해 주어야 한다. 클러스터 개수에 따라 결과값이 완전히 달라지기 때문에 k값의 설정은 매우 중요하다. 따라서 본 논문에서는 k값 설정을 위하여 elbow 기법을 사용하였다.

Elbow 기법이란 K-means 알고리즘의 적정 클러스터 수를 찾아주는 기법으로 오차제곱 합의 값이 최소가 되도록 중심을 결정하는 과정에서 클러스터 개수를 하나씩 늘려 오차제곱의 값이 현저히 작아질 때의 k값을 구하는 방식이다.

2.3 실험 데이터

본 절에서는 실험에 사용된 데이터에 대해 언급한다. 본 논문에서는 본 연구팀이 수집한 비트코인 블록 데이터를 사용하였다. 블록 데이터 중에서 해쉬 값을 제외한 정수데이터들을 중심으로 실험 데이터를 선정하였다. 또한 블록의 높이 200,000부터 560,000 까지의 블록 데이터를 사용하였다.

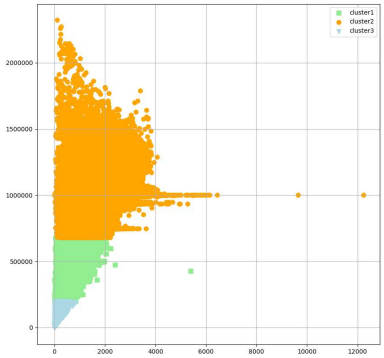
비트코인 블록 데이터	타입
height	Int32
nTx	Int64
size	Int32
nonce	Int32
weight	Int32
Difficuly	Int32
Confirmations	Int32
Strippedsize	Int32

[표 1. 실험데이터]

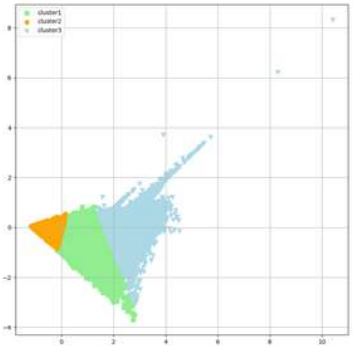
Ⅲ. 실험 및 결과

본 장에서는 PCA를 적용한 K-means 클러스터링 방법을 통해 수집한 비트코인 블록에 대해 실험을 진행한다. PCA를 적용하고 PCA를 통해 축소된 데이터를 활용하여 클러스터링을 진행하였다.

PCA는 2개의 항목(size, nTx) 그리고 36만개의 데이터에 대해 적용하였다. PCA를 적용한 데이터에 대해 K-Means 클러스터링을 적용하였다.



[그림 1. PCA 적용 전 클러스터링 결과]



[그림 2. PCA 적용 2차원 -> 2차원]

2개의 feature를 가지고 실험한 결과 결과들이 조금 더 가독성 있게 나타난다는 것을 발견하였다. 여러 개의 featrue를 가지고 실험을 진행한다면 블록데이터들의 특징을 잘 찾아낼 수 있을 것으로 예상한다.

Ⅳ. 결론 및 향후 연구

본 논문은 비트코인 네트워크에 있어 중요한 항목인 비트코인 블록 데이터의 특징을 분석하기 위한 PCA적용 K-means 클러스터링 방법을 제안하였다. 제안된 방법은 실험을 통하여 타당성을 검증하였다.

향후 연구로는 본 논문에서 실험에 사용한 2가지 feature 이외에 다른 데이터들에 클러스터링 기법을 적용하여 각 클러스터에 속하는 블록들의 특징을 연구 할 계획이다. 또한 같은 방법을 트랜잭션 데이터에도 적용하여 트랜잭션 데이터들의 특징을 분석할 계획이다.

참 고 문 헌

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

[2] J.A. Hartigan,. “Clustering algorithms” (1975).

[3] Jolliffe I.T. “Principal Component Analysis“ (2002)

Named Data Networking with Edge Computing for 5G Radio Access Network

Rehmat Ullah[1], Muhammad Atif Ur Rehman[2], and Byung-Seo Kim[3]

Dept. of Electronics and Computer Engineering, Hongik University[1,2]

Dept. of Software and Communication Engineering, Hongik University[3]

rehmat_ciit@hotmail.com[1], atif_r@outlook.com[2], jsnbs@hongik.ac.kr[3]

Summary

In the upcoming fifth generation (5G) networks, the content distribution and latency will be a very important issue and bring challenges to the existing network infrastructure. In recent, Edge computing and Named Data Networking (NDN) have been introduced as an emerging technology for distributing content closer to end users. The former promises to increase the performance of several applications by using data locality and relieve the core network by addressing the increasing bandwidth demands caused by increased data volume. The latter makes the content directly addressable and routable in networks. It is predicted that if NDN is enabled with Edge computing then it will improve the efficiency of content distribution in 5G networks. Both Edge computing and NDN can improve the communication performance by reducing the distance between users and servers. In this paper, therefore, we enable NDN locally (consumers side) and in Radio Access Network (RAN). We exploit the NDN caching along with Edge computing in order to showcase the pure benefits of NDN over Edge computing for the upcoming 5G networks.

1. Introduction

Initially the Internet was designed to provide communication between end to end hosts. As time goes by, the notion of Internet has been changed due to advancement in technologies such as broadband and mobile devices. Everyday a lot of content is searched and uploaded on the Internet such as YouTube, Facebook, Flicker and Google etc. The content is increasing exponentially, and the multimedia content is forecasted as the majority of the Internet traffic. Therefore, various technologies have been deployed for content dissemination such as content delivery networks (CDNs), peer-to-peer (P2P) networks and Edge computing. The Edge computing is getting more attention due to pushing resources at the network edge; in particular computation, and storage resource are moved closer to end users. Recently the research community come up with a revolutionary Internet paradigm which is known as Information Centric Networking (ICN). Various ICN architectures have been proposed that share common ideas and principles. Due to space issue, the interested readers are referred to [1] for more information. In the ICN paradigm, a unique name is assigned to content, and the content is retrieved without knowing about the location where it resides unlike traditional IP systems. Along with naming the content and devices, it also advocates the content security instead of securing communication channel. Named Date Networking (NDN) [2] is considered as widely used architecture from all of ICN architectures. In this paper, therefore, we enable NDN at the consumer level and evaluate the catching feature of NDN locally (consumers) and at RAN whereas Edge computing is enabled at one hop distance from end users.

The rest of this paper is organized as follows. Section 2 explains the proposed approach. In Section 3, we present the performance evaluation and Section 4 concludes the paper.

2. NDN with Edge Computing

In this paper we show the correlation between cache and network performance using NDN for 5G RAN. In the conventional edge systems, when a user requests some data from the edge server, the first-time data are not found in the edge server. Therefore, the edge server forwards the request to provider for requested data through the core network. The provider replies with content and sends the data back to the edge server. The edge server saves the content. Now in the future, whenever the same content is requested this content can be fetched from the edge server instead the provider. This reduces the latency and traffic for future requests. Although this approach brings the content closer to the user via Edge computing, it has some challenges and this approach still cannot reduce the latency due to core network and extra overhead to the core network. One of the main reasons is that the end users and forwarding routers are not cache enabled. That means if the data is not found at the edge server, then the edge server will forward the requests via core network to the provider. To tackle this issue, recently *He Li et al*. [3] orchestrated NDN at the RAN level, that caches the content on the NDN enabled forwarding devices. Thus, there is no need to access the core network if a cache hit occurs in the RAN. However, NDN is not enabled at the consumer level. In that case every time the user must request the edge server.

We believe that if NDN is enabled at the consumer level along with RAN, then the cache hit ratio could be increased more. As a result, the latency will be reduced as well. In our approach, all the NDN enabled devices communicate with each other at consumer level. If the content is found locally then the user's request will be fulfilled locally. However, if the content is not found locally at the consumer level, then the request will be forwarded to the edge device or base station. If the content is found at the edge node then the content will be forwarded back to user(s). If the content is not found in the edge devices, then the request will be not forwarded to the core network. Instead it will be searched in NDN routers. We use NDN routers in network and NDN protocol is working on the network layer same as [3]. Thus, after receiving the request from the end user device(s), the edge server sends the content request with the NDN protocol. Therefore, it is possible that NDN router(s) in the forwarding route may have cached the required content, and the device(s) can send the content back to the edge server. There is no need to access the core network if a cache hit occurs in the RAN. If the required content is not cached on NDN nodes in RAN, then the core network will be accessed. Enabling NDN at the user level and RAN level is beneficial and the content can be accessed very fast. In our scheme we are boosting the cache hit rate at the end user side and in RAN as well. If more content is pushed closer to users, then high content hit will occur which will reduce the overall latency eventually.

3. Performance evaluation

The choice of the popularity distribution is an important factor that determines the performance of in-network caching in ICN. In our simulation, therefore, we adopted the Mandelbrot-Zipf (MZipf) distribution in ndnSIM [4] to represent content popularity with the settings: p = 5.0, and alpha= 0.7 ~ 1.2. The cache size of each CR is fixed at 100 data objects while we vary the catalogue size from 1000 to 5000 data objects. The total simulation time is 120 seconds. In Figures 1, we show the impact of the cache hit ratio to the catalogue size ratio at local NDN, edge device and NDN enabled forwarding devices. A total of 12000 interest packets are generated. It is evident from Figure 1, that for catalogue size of 3000, 498 number of requests are satisfied from the local NDN and 11502 requests are forwarded to the edge device. Now when requests arrive at the edge device, then 2930 number of requests are satisfied at the edge device, and 8571 number of requests are sent to the NDN enabled forwarding nodes. The forwarding nodes of NDN satisfied 1307 number of requests and 7265 number of requests are sent to the provider i-e Cloud. It is observed that if we enable NDN with the conventional Edge computing systems, the access to the core network will be minimized and the cache hit will occur locally and in the RAN which in turn influences the end to end latency.

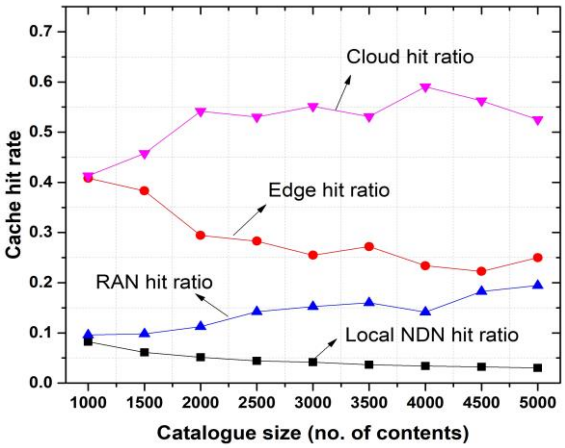


Figure 1: Cache hit rate as a function of catalogue size

4. Conclusion

In this paper we have leveraged NDN for 5G RAN and enabled NDN at the consumer side and RAN as well along with the conventional Edge computing system. Within NDN paradigm, in-network caching is one of the key features and core techniques. Therefore, we have evaluated our work in terms of cache hit ratio which is inversely related to end to end latency. The greater the cache hit ratio; the lower latency will be. Our findings confirmed that enabling NDN at the consumer level is a best choice to reduce the backbone traffic on the edge node(s), forwarding nodes and provider as well.

ACKNOWLEDGMENT

This research was supported in part by the National Research Foundation of Korea (NRF) through the Korea Government (2018R1A2B6002399) and in part by the International Research & Development Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT. (No. NRF-2018K1A3A1A39086819).

References

[1] G. Xylomenos, C. N. Ververidis, et al., "A Survey of information-centric networking research," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.

[2] Jacobson, V., Smetters, D., Thornton, J., et al. "Networking named content", *Proceedings of the 5th international conference on Emerging networking experiments and technologies.* ACM 2009 pp. 1-12.

[3] H. Li, K. Ota and M. Dong, "ECCN: Orchestration of Edge-Centric Computing and Content-Centric Networking in the 5G Radio Access Network," *in IEEE Wireless Communications*, vol. 25, no. 3, pp. 88-93, JUNE 2018.

[4] S. Mastorakis, A. Afanasyev, and L. Zhang, "On the Evolution of ndnSIM: an Open-Source Simulator for NDN Experimentation," *ACM Computer Communication Review, July 2017.*

혼잡한 대중교통 상황에서의 p2p offloading 기법 연구

백호성, 백상헌*
고려대학교
{gh1emd, shpack}@korea.ac.kr

A study on the p2p task offloading scheme in dense transportation environment

Ho Sung Baek, Sang Heon Pack*
Korea Univ.

요 약

본 논문에서는 혼잡한 대중교통 상황에서의 p2p offloading 방안을 연구하였고, 어떠한 요소들이 고려되어야 하는지에 대해 분석하였다. 스마트폰, 태블릿과 같이 mobile device가 대중적으로 보급화되면서, mobile device 위에서 다양한 application들이 사용되고 있다. 특히, 최근에는 AR/VR과 같이 높은 연산능력을 요구하는 application들이 개발되고 있고, 점차적으로 mobile device에서 이러한 application들이 도입되고 있다. 하지만, 제한된 연산능력과 배터리 용량을 지닌 mobile device에서는 높은 연산능력을 요구하고 배터리 소모가 큰 application을 사용하는 것이 제한될 수 있다. 따라서, 본 논문에서는 위와 같은 문제를 해결하기 위해, 혼잡한 교통상황에 존재하는 다른 mobile device들에게 task를 offloading하는 방안을 제시하고, 이 경우에 고려되어야 하는 요소들에 대해 분석하고 연구방향을 제시한다.

Ⅰ. 서 론

기술이 발전됨에 따라, 스마트폰, 태블릿과 같은 mobile device는 이제 언제 어디서나 다수의 사람이 한 대 이상씩 지니고 있을만큼 널리 보급되었다. 이에 따라, mobile device 위에서 다양한 application들이 개발되어 사용되고 있고, AR/VR과 같은 높은 연산 능력을 요구하는 application들이 출시되고 있다. 하지만 mobile device는 한정된 연산 능력을 갖고 있고, 휴대용 기기의 특성상 배터리 소모의 문제에 따라 연산 능력이 제한될 수 있다. 이와 같은 문제를 해결하기 위해, 본 논문에서는 혼잡한 교통상황에서의 p2p offloading 기법을 제시한다.

혼잡한 대중교통 상황에서는 mobile device를 소유한 여러 개인이 밀집되어 있고, 일정 시간 동안 연산 능력을 지닌 여러 mobile device가 같은 공간 내에 존재한다. 즉, 제한된 배터리 용량과 연산 능력을 지닌 mobile device는 주변에 존재하는 mobile device들에 자신이 처리해야 할 일부 혹은 전체 task를 offloading 시킬 수 있다. 하지만 task를 offloading 받은 다른 mobile device들이 이에 대해 협조적으로 하기 위해서는 적절한 incentive가 필요하다. 또한 central authority가 없는 p2p 환경에서 offloading을 시키기 위해서는 신뢰적인 offloading 방안이 필요하다. 따라서 본 논문에서는 혼잡한 대중교통 환경에서의 p2p offloading 기법을 제시하고, 이 경우 고려되어야 하는 요소들을 분석한 뒤, 결론에서는 향후 연구 방향에 대해 제시한다.

Ⅱ. 본 론

그림 1은 본 논문에서 제안하는 혼잡한 교통상황에서의 p2p offloading 예제를 나타낸다. offloading 하는 절차는 다음과 같다. mobile device의 배터리가 충분하지 않은 상황에서 AR/VR과 같이 높은 연산 능력을 요구하는 application을 사용하고자 하는 사용자는 전력 소모가 큰 여러 task를 주변 mobile device에게 offloading 시키려고 한다. 따라서 현재 자신이 위치한 대중교통 내에 p2p network에 참여하고 있는 peer list를 서버로부터 받고, 해당 peer list에 task를 offloading 시키려고 한다는 메시지를 브로드캐스팅한다. 이를 받은 여러 mobile device 중에 task offloading을 받고, 이를 처리하여 incentive를 받으려 하는 mobile device는 자신의 잔여 computing power를 포함하여 회신을 한다. 이를 받은 사용자는 알맞은 기준에 따라 offloading 시킬 mobile device를 선택하고 task를 offloading 시킨다. offloading을 받은 mobile device는 이를 처리하고 처리된 결과를 다시 사용자에게 보내면, 사용자는 확인 후에 task의 비중을 고려하여 incentive를 부여하고 offloading 절차는 종료된다. 이를 통해, 사용자는 적은 양의 전력 소모를 통해 AR/VR과 같은 서비스를 제공받을 수 있고 낮은 연산 능력을 지닌 mobile device도 AR/VR과 같은 application을 사용할 수 있다.

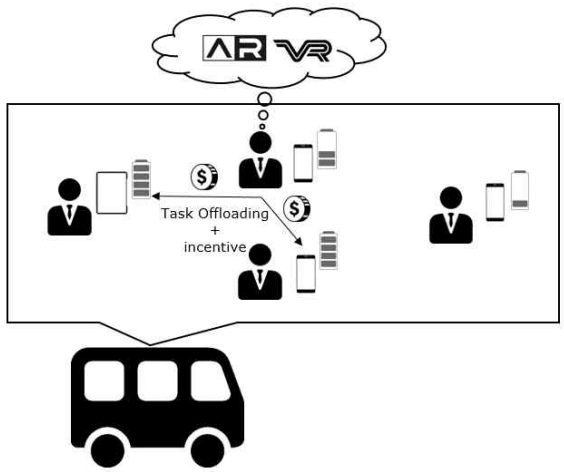


그림 1. 대중교통에서의 p2p offloading 예제

위에서 설명한 offloading 절차에서 고려되어야 하는 요소들은 다음과 같다. 우선 p2p network에서는 offloading을 시키고, 처리된 결과가 잘 처리되었는지 확인하는 central authority가 존재하지 않는다. 즉, 이로 인해 다음과 같은 문제들이 발생할 수 있다. mobile device들은 서로에 대해 알 수 없고 신뢰성이 떨어진다. 따라서 offloading을 받은 mobile device들은 incentive를 받기 위해, 제대로 처리하지 않은 결과값을 보내고 incentive를 받는 경우가 발생할 수 있다. 따라서 처리된 결과값을 확인하는 방안을 추가되어야 한다. 또한 offloading 분야에서 항상 문제가 제기되고 있는 사용자의 usage pattern과 같은 privacy 측면도 고려되어야 한다. 일정시간 동안 하나의 mobile device에게 모든 task를 offloading 하면, 이를 이용하여 사용자의 data usage pattern과 같은 privacy가 침해될 수 있기 때문이다.

이와 같은 요소들을 고려하기 위해 연구를 다음과 같이 진행하려 한다. 처리된 결과값을 확인하는 방안으로, offloading 시킬 task 중 일부의 task를 서로 다른 mobile device에게, 혹은 사용자의 mobile device에 중복적으로 할당한다. 즉, 총 task가 {task1, task2, ... , task7} 7개 있을 경우, mobile device 1,2에게 각각 {task1, task2, task3, task4}, {task4, task5, task6, task7}을 할당한다. task 4가 중복적으로 할당된다. offloading 받은 mobile device 들은 할당받은 task를 처리 후에, 각각의 task에 해당하는 결과값을 hash function을 돌려 각각의 hash값과 결과값을 사용자에게 전송한다. 이를 받은 사용자는 중복할당한 task의 hash 값을 비교 후에 값이 같으면, 잘 처리되었다고 판단하고 incentive를 부여한다. 이 경우, mobile device들은 어떤 task가 중복적으로 할당되었는지 모르기 때문에, 모든 task를 잘 처리하고 이를 전송했을 시에 받을 incentive와 확률적으로 일부의 task를 처리하고 전송하여 받을 수 있는 incentive의 기댓값을 비교하여 자신의 cost 대비 incentive가 높다면 모든 task에 대해 잘 처리할 것이다. privacy 측면에서 data usage pattern은 사용자가 하나의 device가 아닌 두 개이상의 mobile device에게 task를 할당한다면 일정 부분 해결될 수 있다. 일부의 task를 통해 사용자의 usage pattern을 알기는 어렵기 때문이다. 또한 사용자가 task를 모두 처리하고 또 다른 task를 처리할 경우, 이전의 선택한 mobile device를 최대한 선택하지 않는 알고리즘을 통해 privacy 측면은 더 보완될 수 있다.

Ⅲ. 결 론

본 논문에서는 혼잡한 대중교통에서의 p2p offloading 기법에 대해 제안하였고, 이 경우 고려되어야 하는 요소들과 향후 연구 방향에 대해 제시하였다. 기술의 발전으로, 스마트폰이 대중적으로 보급됨에 따라 이를 잘 활용할 수 있는 방법을 모색하면 다양한 분야에서 활용될 수 있을 것으로 보인다. 향후 연구로는 제안 기법을 구체화시키고, 제안 기법의 타당성을 수학적으로 증명할 것이다. 또한 [1]에서의 reputation mechanism을 활용하여 보다 더 효율적으로 mobile device를 선택하고 offloading을 시킬 예정이다.

ACKNOWLEDGMENT

이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 (No.2015-0-00575, 글로벌 SDN/NFV 공개소프트웨어 핵심모듈/기능 개발) 지원과 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단 (전략과제)의 지원을 받아 수행된 연구임(No. 2017R1E1A1A01073742)

참 고 문 헌

[1] D. Chatzopoulos, M. Ahmadi, S. Kosta, and P. Hui, “FlopCoin: A Cryptocurrency for Computation Offloading,” IEEE Transactions on Mobile Computing, Vol. 17, No. 5, pp. 1062-1075, May 2018.

허가형 블록체인을 이용한 마이크로 그리드 에너지 공유 프레임워크

전정민 ˚, 강선무, 홍충선*

경희대학교

jmjeon0212@khu.ac.kr ˚, etxkang@khu.ac.kr, *cshong@khu.ac.kr

MicroGrid Energy Sharing Framework using Permissioned Blockchain

Jeongmin Jeon ˚, Sunmoo Kang, Choong Seon Hong*

*Kyung Hee University

요 약

미래사회는 에너지인터넷을 구축하여 에너지를 공유하고 더 나아가 에너지 사용과 온실가스 사용을 최소화하는 제로 에너지 지향사회로 나아갈 것이다. 이에 따라 에너지 시스템은 중앙 집중형 에너지 공급 시스템에서 ICT기반 분산형 에너지 공급 시스템으로 패러다임이 변화하고 있다. 이러한 변화 트렌드에 적용될 수 있는 ICT융합 기법으로써, 본 논문에서는 Permissioned Blockchain 기반 스마트 컨트렉트를 활용한 에너지 공유 프레임워크를 제안한다. 이는 에너지 공유 시 중앙화된 회사나 데이터베이스가 거래 데이터를 참조하는 경우 발생하는 데이터 위·변조에 대한 위험성, 다시 말해 데이터 무결성이 보장되지 않는 문제를 해쉬 기반 Winternitz One-Time Signature(W-OTS) 기법을 적용하여 해결함으로써 중개자 개입 없이 가까운 거리에 있는 프로슈머와 소비자 사이에서 에너지를 안전하게 공유할 수 있는 환경을 제공한다.

1. 서 론

최근 SmartGrid(SG)에서 개인 간의 전력거래를 위한 블록체인 적용하려는 움직임이 활발하다. 전력망과 통신기술의 융합으로 탄생한 전력 데이터 거래 시스템은 기존 전력망과 비교하면 안정성, 효율성을 가진다. 신재생 에너지, 전기자동차, 프로슈머 등 새로운 서비스가 등장하여도 전력망과의 유연하게 연동할 수 있는 확장성을 향상한다. SG와 블록체인 기술의 결합은 프로슈머와 소비자의 전력거래를 할 수 있는 전력망으로 진화해오고 있다[1].

현재 우리나라는 프로슈머와 소비자 간의 전력을 거래할 수 있는 서비스가 필요하지만, 현재 이와 관련된 안전한 전력거래 인프라가 국내에 존재하지 않은 실정이다[2]. 일반적으로 에너지 요금부과 수익은 데이터 또는 서비스에 따라 달라지기 때문에 데이터 무결성은 에너지 거래 사기 문제를 예방하는데 핵심적인 역할을 한다[3].

4차 산업혁명과 함께 스마트 미터를 이용해 전력 사용정보를 공급자에게 제공하는 SG 시스템과 한 단계 발전하여 에너지의 소비와 생산 및 판매를 같이하는 프로슈머(Prosumer)의 개념이 확대된 MicroGrid(MG) 시스템이 등장하였다[4]. MG와 블록체인이 제공하는 장점은 데이터의 무결성,신뢰성이 보장되며 중앙으로 모여있지 않고 데이터의 분산된 분권화 및 보안, 투명성을 지킬 수 있다.

따라서 본 논문에서는 Permissioned Blockchain(PB)을 기반으로 스마트 컨트렉트를 활용한 보안성이 높고 투명한 전력을 거래가 가능한 MG 에너지 공유 프레임워크를 제안한다. 이때 중앙화된 회사나 Database(DB)에 의해 위변조를 방지하기 위해 IOTA에서는 W-OTS를 사용한다. 이를 블록체인과 분산된 복잡한 에너지 거래 및 데이터 교환의 무결성과 신뢰성 확보와 관련된 복잡한 문제를 해결하고자 한다. 스마트 컨트랙트는 에너지 공급자와 소비자 간의 미리 정의된 규칙을 기반으로 감시 가능한 다중 상대 트랜잭션을 지원하는 자동화된 스마트 컨트렉트을 가능하게 한다[3].

본 논문의 2장에서는 블록체인에 개념, 해쉬 기반 서명 기법 W-OTS 를 살펴보고, 3장에서는 본 논문의 프레임워크 시스템 모델과 work flow 다이어그램에 관한 내용을 설명한다. 4장에서는 기존에 비트코인에 서명검증인 ECDSA(Elliptic Curve Digital Signature Algorithm)과 무결성 검증을 위해 제안하는 W-OTS 내용과의 서명 알고리즘 서명 비교 내용을 다루며, 마지막 5장은 본 논문의 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

2.1 블록체인(Blockchain)의 개념

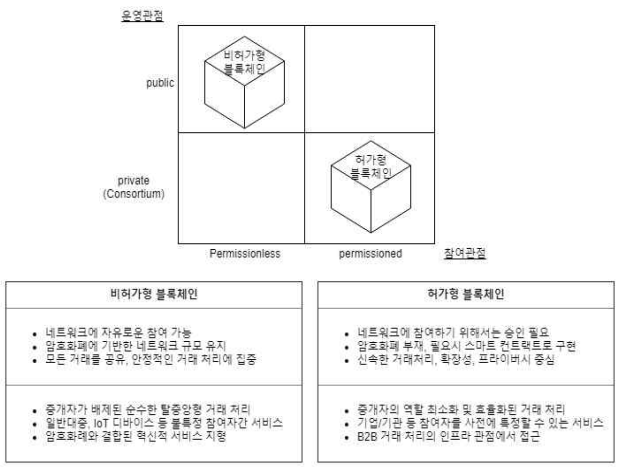


그림 1. 블록체인 유형별 특징

블록체인 기술은 변조 및 수정으로부터 보호된 블록이라고 불리는 지속적으로 증가하는 레코드 목록을 유지 관리하는 분산 데이터베이스이다. 각 블록은 타임 스탬프와 이전 블록에 대한 링크를 포함한다[5]. 스마트 컨트렉트은 가치를 교환하는 기술이나 응용프로그램으로 정의된다. 허가형(permissioned) 블록체인은 합의과정에 참여하려면 사전 승인이 필요하며, 참여자 개개인을 지정하는 프라이빗(private)블록체인과 특정 그룹 내에 사전 합의에 따라 쓰기 권한을 가지는 컨소시엄(consortium)블록체인으로 분류된다[6]. 본 논문에서는 기존 중앙화된 전력시스템에 문제점을 MG에서 허가형 블록체인으로 해결하면서 프로슈머와 소비자 간에 안전하고 데이터의 무결성이 보장되는 에너지 계량 및 청구 시스템을 단순화할 수 있는 프레임워크를 제안한다. 또한 허가형 블록체인에 트랜잭션 검증방법과 PB에 최적화된 W-OTS 기법을 비교하고 성능 측면에서의 PB 기반 에너지 공유 프레임워크의 이점을 보이고자 한다.

2.2 해쉬 기반 서명 기법 W-OTS

기존에 서명기법에 효과적인 서명 생성뿐만 아니라, 서명의 크기가 서명키, 검증키와 같이 매우크다는 단점을 보완을 위해 메시지 다이제스트값의 일부분의 비트에 대해 동시에 서명하는 방식을 사용하여, 서명키,검증키, 서명의 종합적으로 압축시켰다[7]. 분산원장에서 사용되는 Merkle의 트리 인증과 같이 사용하는게 적합하다.

이는 Merkle Signature scheme(MSS)의 크기를 확연하게 줄인다. 따라서 W-OTS는 효과적인 MSS를 가능하게 하며 센서 네트워크 및 분산원장 프로토콜에서 인증에 사용된다.

본 논문에서는 중앙화된 회사나 DB에서 참조하게 될 때 데이터 무결성을 보장하기 위해 W-OTS 방법을 통해 위변조 방지를 제안한다.

3. 제안사항

3.1 구조

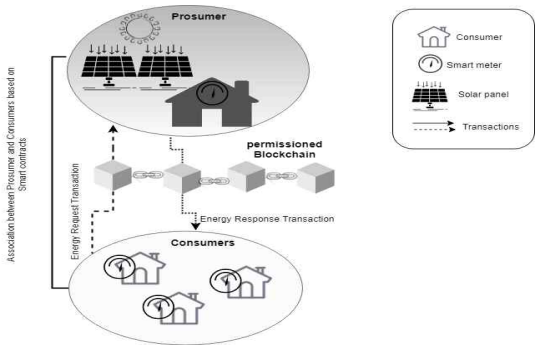


그림 2. System Model

본 논문은 그림 2과 같이 MG에서 허가형 블록체인을 활용한다. 스마트 미터 장치는 전기 에너지를 판매하고 구매할 수 있는 자동화된 에이전트 역할을 할 수 있다.

본 논문에서 제안하는 시스템 모델을 활용하여 에너지가 필요로 하는 소비자가 프로슈머에게 에너지를 요청을 하면 중개자 없이 허가형 블록체인 스마트 컨트렉트를 통하여 거래 내역이 저장이 된다. 블록이 체인에 저장되고 소비 전력을 추가 또는 제거하려면 스마트 컨트렉트을 제거하여 스마트 컨트렉트을 업데이트 해야하다[8]. 블록체인 네트워크 내의 참가자가 특정 제작자가 두 번 판매하지 않고 전기 에너지를 실제로 그리드에 주입 했는지 여부를 서로 독립적으로 확인할 수 있다.

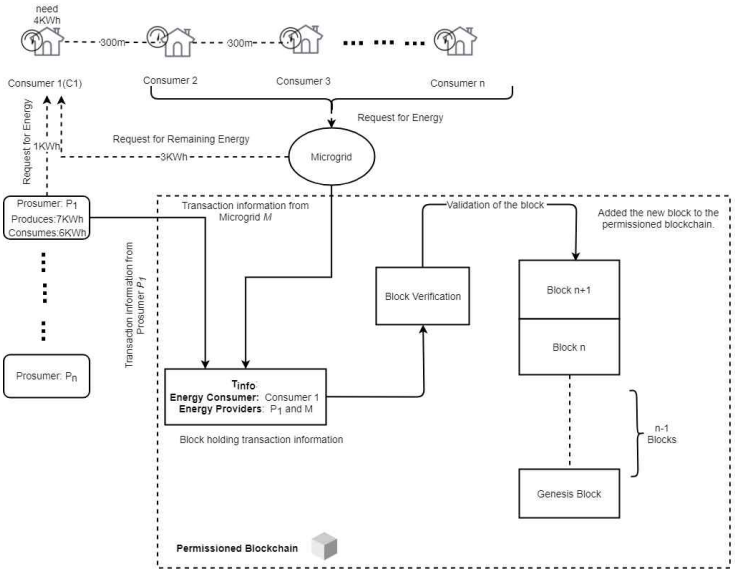


그림 3 Work-Flow Diagram

그림 3는 [8]에서 허가형 코인 기반에 스마트컨트랙트 에너지 거래를 활용하여 MG에서 PB 기반에 스마트 컨트렉트를 이용한 거래를 통해 기밀성, 무결성, 유효성을 개선한 프레임워크를 제안한다. 그림3은 Work-Flow Diagram이며, 스마트 미터는 중개자 대신에 전기 에너지를 거래할 수 있는 자동화된 Agent 역할을 한다. N개에 Smart meter 인프라가 구축된 소비자는 필요한 KWh를 독립된 분산전원으로 국소적인 전력공급 시스템인 MG[1]를 통해서 여분에 신재생 에너지 또는 태양열 에너지가 있는 프로슈머에게 요청하고 거래할 수 있다.

Micro grid Level:

① “C1” 은 “P1” 스마트 컨트랙트에 따라 4KWh를 요청한다.

② MG는 전기 라우팅 및 전력선 지원을 기반으로 트랜잭션을 수행 할 수 있는지 여부를 확인한다.

③라우팅이 유효화되면 MG는 “P1” 이 거래를 지원하는 회선 용량과 관련하여 “C1” 에게 전송할수 있게 한다.

④ MG는 거래조건을 충족 시키기 위해 MG에게 잔여 전

력을 요청한다.

⑤ 네트워크는 P1에 잔여전력 1KWh를 MG를 통해 C1에게 제공한다.

⑥ 마지막으로 , MG는 3KWh를 “C1”로 보낸다.

4. 성능 및 속성 비교

그림 4 트랜잭션 서명검증 결과 그래프

표 1 트랜잭션 서명 검증 (단위:ms)

알고리즘		구분
ECDSA		2.048
W-OTS	w=2	0.435
	w=3	0.537
	w=4	0.603

본 검증에서는 트랜잭션에 대해 검토할 때 임의의 트랜잭션에 대해 ECDSA 서명을 만들고 그 서명을 검토하는데 걸리는 시간과 Merkle 서명에 대한 서명 검증에 걸리는 시간을 서로 비교하는 실험을 하였다. 기존 시스템과 비교했을 때, 기존 블록체인 트랜잭션에 대해 검증으로는 한번만 검증하지만, W-OTS를 활용하게 되면 첫 번째 단계로 트랜잭션 서명에 대한 검증을 수행하고 2단계로 검증키와 인증 경로의 값들을 이용해 머클 트리 값을 구하여 한번 더 검증하는 방식이다. 단계적인 측면에서는 기존 블록체인이 속도 면에서 우위일것으로 예상되었으나, W-OTS기법이 4-5배 빠르게 서명에 대해서 검증한다는 사실을 확인할 수 있다. 뿐만 아니라 서명 검증 시 MSS를 이용함으로써, 기존의 ECDSA 서명 검증과는 다르게 추가로 머클 루트의 값에 대한 무결성을 검증할 수 있었다[9]. 본 논문에서는 중앙화된 기관이나 데이터베이스가 참조할 때 W-OTS 기법을 사용하여 에너지 거래 시 생기는 트랜잭션에 보다 신속하고 안전하게 무결성을 보장할 수 있음을 보였다.

5. 결론 및 향후 연구

본 논문에서는 W-OTS를 통해 PB에서 거래내역을 재참조를 할 때 발생할 수 있는 데이터의 위·변조를 막고 데이터의 무결성을 지켰다.

스마트 컨트렉트를 통해 가까운 거리 내에 있는 소비자는 MG를 통하여 프로슈머에게 원하는 에너지를 요청을 하면 안전하게 지역 내 에너지를 공유할 수 있음을 확인했다. PB 블록체인을 기반으로 하였기 때문에 소비자,프로슈머, MG 및 정부기관(KEPCO)가 데이터 공유 및 처리를 안전하게 할 수 있는 이점을 보였다.

향후 연구로는 Hybrid Blockchain(BC)를 사용하여 에너지 이중 지불과 같은 문제점을 해결하는 것을 연구할 계획이다.

ACKNOWLEDGMENT

본 연구는 산업통상자원부(MOTIE)와 한국에너지기술평가원(KETEP)의 지원을 받아 수행한 연구입니다. (No. 70300038) *Dr. CS Hong is the corresponding author

6. 참고문헌

[1] 홍원표,(2018).최근 마이크로그리드 기술개발 동향 분석.조명・전기설비,32(6),40-52.

[2] Winter, Thomas. “The Advantages and Challenges of the Blockchain for Smart Grids.“ (2018).

[3] Sabah Suhail, Choong Seon Hong, M Ali Lodhi, Faheem Zafar, Abid Khan, and Faisal Bashir. Data trustworthiness in IoT. In Information Networking (ICOIN), 2018 International Conference on, pages 414-419. IEEE, 2018.

[4] 박찬국, 김양수,(2016).우리나라 P2P 전력거래 가능성 연구. 에너지경제연구원 수시연구보고서,1-85,2016

[5] L. Trottier, “original-bitcoin”, 2013, [Online]. Available on Github

[6] P. Franco, “Understanding Bitcoin: Cryptography, Engineering and Economics”, John Wiley & Sons. p. 9, 2014

[7] Billet, O., Robshaw, M. J., Peyrin, T. (2007, July). ” On building hash functions from multivariate quadratic equations” In Australasian Conference on Information Security and Privacy pp. 82-95.

[8] Nehai, Zeinab & Guérard, Guillaume. (2017). INTEGRATION OF THE BLOCKCHAIN IN A SMART GRID MODEL. CYSENI. 2017.

[9] 배봉진, “블록체인에 대한 해시 기반 서명 기법 적용방안”.부산대학교 대학원 석사학위논문, 2017.

합의 알고리즘 성능 검증을 위한 블록체인 네트워크 시뮬레이터 모델

강창훈, 고경찬, 홍원기
포항공과대학교 컴퓨터공학과

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

Blockchain Network Simulator Model for Verifying Performance of Consensus Algorithms

Changhoon Kang, Kyungchan Ko, James Won-Ki Hong
Department of Computer Science and Engineering, POSTECH

요 약

블록체인은 분산된 여러 개의 Node 들이 모두 동일한 데이터 사슬을 저장 및 관리하도록 하여 누구도 데이터를 임의로 수정할 수 없으면서 이를 투명하게 관리하는 기술이다. 이때 블록체인 네트워크상에서의 합의 알고리즘은 모든 Node 들이 동일한 의사결정을 통하여 똑같은 데이터를 갖도록 하기 위해 사용된다. 블록체인에 대한 관심이 높아지고 이를 도입하는 것을 통해 기존에 존재하던 여러 이슈들을 해결하려는 시도가 많아지면서[1], 각 사례에 맞는 합의 알고리즘들 역시 다양하게 제안되고 있다. 합의 알고리즘에 따라 이를 적용한 블록체인이 갖는 TPS(Transaction per Second), 블록 생성 주기, 각 거래의 완결성(Finality) 등 성능과 관계된 특징들이 달라진다. 제안된 합의 알고리즘이 어떤 성능을 갖는지, 충분한 보안 수준을 보장하는지에 대한 수학적인 검증들은 각 블록체인의 백서에서 제공되고 있다. 하지만 이것이 실제 블록체인 네트워크 환경에서 어떻게 동작하는지 확인하는 실험적인 검증은 많은 비용과 시간 소요 등의 문제로 잘 이루어지지 않고 있다. 일단 한 번 실제 환경에 합의 알고리즘을 구현하고 나면 추후 변경하는 것에 굉장히 많은 시간과 비용이 소모되기 때문에 우선적으로 실험적인 검증을 거치는 것이 필요하다. 본 논문에서는 가상의 블록체인 네트워크를 구성하고 제안된 합의 알고리즘을 적용시켜 성능을 검증할 수 있는 시뮬레이터 모델을 제안한다. 합의 알고리즘을 몇 가지 Parameter 들의 설정을 통해 표현할 수 있도록 하는 방법을 구현하고, 비교적 간단하고 적은 비용으로 합의 알고리즘의 실험적인 검증을 가능하게 하는 모델을 제안하는 것을 목표로 한다.

Ⅰ. 서 론

블록체인의 합의 알고리즘은 블록체인 네트워크 상의 많은 Node 들이 모두 동일한 의사결정 과정을 통해 같은 데이터를 저장하고 관리하도록 해주는 역할을 한다. 블록체인 상에서 매번 오직 하나의 동일한 블록만이 각 Node 가 갖고 있는 사슬에 추가되게 된다. 이 과정에서 만약 여러 Miner 가 모두 블록 생성에 성공한 경우, 블록을 수신한 각 Node 들이 어떤 블록을 추가할지 정한다. 또한 네트워크상의 Propagation Delay 등의 이유로 같은 높이에 서로 다른 블록이 추가되는 fork 가 발생한 경우, Longest Chain 을 선택하게 만듦으로써 다시 동일한 체인을 갖도록 해결해주는 역할도 수행한다.

블록체인의 성능을 논할 때 주로 해당 블록체인의 초당 최대 거래 처리 건수(TPS), 블록의 생성 주기, 각 거래가 완결될 때까지 필요한 시간 등을 고려한다. 각각의 블록체인이 사용되는 목적에 따라 거래 속도와 보안 성능 관계처럼 트레이드오프 관계에 있는 특징들의 중요도를 적절히 분배한 새로운 합의 알고리즘들이 제안되기도 한다. 백서에서 수학적인 검증을 통해 새롭게 고안된 합의 알고리즘이 어느 정도의 보안 수준과 성능을 갖는지 분석하는 검증을 거치고, 실제 블록체인 환경을 구축할 때 적용된다. 하지만 실제 환경상에서 기존의 예측과는 다르거나 본래 목표했던 성능에 미치지 못하는 경우 이를 다시 변경하는 것에 있어 매우 많은 시간과 비용이 소모된다. 따라서 이를 예방하기 위해 실제 적용 전 미리 수학적 검증과 더불어 실험적인 검증까지 거치는 것이 중요하다. 하지만 해당 실험 환경을 구성하는 것 역시 어려움이 따른다.

본 논문에서는 하나의 컴퓨터 상에서 시뮬레이터를 통해 가상의 블록체인 네트워크 환경을 구성하고, 합의 알고리즘을 표현, 적용시킬 수 있는 방법을 통해 비교적 간단하고 적은 비용으로 합의 알고리즘의 실험적인 검증이 가능하도록 하는 모델을 제안하는 것을 목표로 한다.

Ⅱ. 관련 연구

본 논문에서 목표로 하는 시뮬레이터보다 비교적 간단한 오픈 소스 프로젝트들이 많이 진행되고 있었다. 하지만 대부분 각 시뮬레이터 당 기존에

존재하는 하나의 합의 알고리즘을 초점으로 하는 시뮬레이션이 구현되어 있었다.

논문 [2]의 저자는 블록체인 네트워크 시뮬레이터를 개발하여 블록체인 네트워크 자체의 성능을 검증하는 것을 주요 목적으로 한 연구를 진행하였다. 다양한 Parameter 들을 조정하는 방법을 도입하여 사용자가 원하는 네트워크 및 합의 알고리즘 환경을 구성하고 이를 테스트 해볼 수 있도록 하였다. 해당 시뮬레이터를 사용자들이 사용할 때, 네트워크를 구성하는 부분에 있어서는 직관적인 Parameter 들을 통해 간단하고 쉽게 디자인되었지만, 합의 알고리즘을 적용하는 과정에 있어서는 다소 사용하기 어렵다. 그리고 확률 개념을 도입하여 사용자가 직접 합의 알고리즘에 따라 적절한 설정을 해주어야 한다. 또한 이번 연구에서 초점을 두고 있는 합의 알고리즘 자체의 검증과는 거리가 있는 연구라고 생각된다. Parameter 들의 설정을 통해 사용자가 직접 원하는 환경을 구성하는 방식을 통해 사용자가 직접 새롭게 고안해낸 합의 알고리즘을 시뮬레이터 상에 구현할 수 있도록 한다는 아이디어를 이 논문으로부터 얻었다.

논문 [3]은 IBM Hyperledger Fabric 에서 설정 Parameter 들을 변경해가며, Transaction 의 Throughput(TPS)과 Latency 측정을 통해 그것의 성능을 특정하고 있다. 또한 이를 기반으로 시스템 상에서 병목현상이 발생하는 지점을 찾아내어 성능을 더욱 높일 수 있는 설정 가이드라인을 제시한다.

Ⅲ. 시뮬레이터 모델 디자인

1. 네트워크 구성

합의 알고리즘의 성능 검증을 위해 이를 적용시킬 블록체인 네트워크를 구성하는 것이 필요하다. 가상 네트워크를 구성하는 단계는 본 논문에서 주요하게 고려하는 합의 알고리즘을 가상 네트워크에 적용하는 것과 거리가 먼 내용이기 때문에 기존의 네트워크 Emulator 를 사용하는 것이 효율적이다. 그래서 Python API 를 제공하는 Mininet 을 사용하는 것을 통해 가상의 네트워크를 구축할 수 있는 방법을 사용한다.

Mininet 을 이용하면 사용자가 직접 여러 명령어를 통해 본인이 원하는 네트워크 환경을 구성할 수 있다. 다만 본 논문에서 제안하는 모델을 이용할 사용자들이 Mininet 의 사용법까지 익혀야 하는 부담이 생기게 된다. 따라서 이를 덜기 위해 사용자로부터 간단한 옵션들을 입력 받고 이를 토대로 시뮬레이터가 직접 가상 네트워크를 구성할 수 있도록 한다. 예를 들면 사용자가 Mininet 의 명령어를 알 필요 없이 네트워크 구성에 필요한 최소한의 정보들을 숫자나 간단한 옵션을 통해 전달할 수 있도록 한다. 또한 일반적인 네트워크 환경 역시 미리 제공하여 특별한 네트워크 환경상에서의 실험이 필요한 경우가 아니라면 따로 직접 네트워크를 구성할 필요 없이 이를 사용할 수 있도록 한다.

2. 합의 알고리즘 적용

합의 알고리즘들은 각각 서로 다른 특징들과 동작 과정을 갖고 있다. 따라서 모든 합의 알고리즘들을 다 적용 가능하도록 만들기 위해서는 이들을 모두 표현할 수 있는 일반적인 Parameter 들을 추출하는 것이 필요하다. 사용자들은 이렇게 제공되는 Parameter 들을 자신이 실험하고자 하는 합의 알고리즘에 알맞게 설정하는 과정을 통해 위에서 만든 가상 네트워크 상에 합의 알고리즘을 적용시킨다.

합의 알고리즘이 제공하는 기본적인 기능인 새롭게 추가될 하나의 블록을 고르는 것은 논문 [2]에서 도입하였던 확률 개념을 사용한다. 적용시키려는 합의 알고리즘에서 블록이 선택되는 기준에 따라 각 Node 들이 채굴을 성공할 수 있는 확률을 다르게 설정해주고 실험 중에도 상황에 따라 유동적으로 변동될 수 있도록 한다. 그 외 블록이 생성되는 과정이나 실제로 Validation 에 참여할 Node 들을 선택하는 것, DPoS 에서의 Voting 과 같이 추가적으로 필요한 다른 프로세스가 있다면 이 역시 추가할 수 있도록 한다.

3. 성능 검증

논문 [3]과 비슷하게 TPS, 블록 생성 주기, 각 거래가 완결될 때까지의 시간을 측정한다. 시뮬레이션 하는 동안 지속적으로 해당 값들을 측정하여 제공한다.

Ⅳ. 향후 연구 과제 및 결론

향후에는 본 논문에서 제안한 모델을 직접 구현하여 시뮬레이터를 개발하고, 테스트하는 과정이 필요하다. 기존에 존재하던 한정된 합의 알고리즘에 대해 시뮬레이션이 가능한 이미 검증된 시뮬레이터와 해당 알고리즘을 구현한 본 논문의 시뮬레이터 실험 결과를 비교하는 것을 통해 본 모델의 성능을 알아보아야 한다.

본 논문의 모델은 모든 종류의 합의 알고리즘들을 구현하고 실험해볼 수 있다는 점에 있어서 다른 기존의 시뮬레이터들과 차별성이 있다. 개발 과정을 통해 조금 더 자세한 부분들까지 모델에 추가되어야 할 것이라 판단된다.

ACKNOWLEDGMENT

이 논문은 2019 년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구 임 (No.2018-0-00539)

참 고 문 헌

[1] Ferrag, Mohamed Amine, et al. "Blockchain technologies for the internet of things: Research issues and challenges." IEEE Internet of Things Journal (2018).

[2] Aoki, Yusuke, et al. "SimBlock: A Blockchain Network Simulator." arXiv preprint arXiv:1901.09777 (2019).

[3] Thakkar, Parth, Senthil Nathan, and Balaji Viswanathan. "Performance benchmarking and optimizing hyperledger fabric blockchain platform." 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS). IEEE, 2018.

머신 러닝을 이용한 비트코인 트랜잭션 수수료 예측

최원석, 고경찬, 홍원기
포항공과대학교 컴퓨터공학과

{ws4583, kkc90, jwkhong}@postech.ac.kr

Bitcoin Transaction Fee Estimation Using Machine Learning

Wonseok Choi, Kyungchan Ko, James Won-Ki Hong
Department of Computer Science and Engineering, POSTECH

요 약

비트코인에서는 트랜잭션을 생성할 때 사용자가 지불할 트랜잭션 수수료를 미리 정해야 하고 수수료에 따라 트랜잭션이 확정되기까지의 시간이 달라진다. 일반적으로는 더 많은 수수료를 지불할수록 소요 시간이 짧아지게 되나 사용자는 일정 시간 이내에만 트랜잭션이 확정되면 되므로 굳이 높은 수수료를 지불할 필요가 없고, 수수료에 따른 정확한 소요 시간도 알 수 없다. 따라서, 사용자의 입장에서는 트랜잭션 수수료를 최소화 하기 위해 트랜잭션 수수료에 따른 소요 시간을 예측할 수 있는 모델이 필요하다. 현재 다양한 트랜잭션 수수료 예측 모델이 존재하지만 비트코인 트랜잭션 수가 시시각각으로 급변하기 때문에 정확한 트랜잭션 수수료 예측 모델을 만드는 일은 쉽지 않다. 본 논문에서는 이를 위해 트랜잭션 수수료 예측 모델을 제작하기 위해 머신 러닝을 기술을 도입하였다. 본 논문에서 사용한 방법은 트랜잭션이 Mempool 에 들어왔을 때 Mempool 에 존재하는 트랜잭션들의 수를 고려하여 비트코인 트랜잭션들의 트랜잭션 수수료와 트랜잭션들이 확정되기까지의 시간과의 관계를 학습시키는 것으로 이 방법을 통해 비트코인 트랜잭션 수수료 예측 모델을 만드는 것을 목표로 한다.

Ⅰ. 서 론

비트코인에서는 트랜잭션을 생성할 때 사용자가 지불할 트랜잭션 수수료를 미리 정해야 하고 수수료에 따라 트랜잭션이 확정되기까지의 시간이 달라진다. 일반적으로는 더 많은 수수료를 지불할수록 소요 시간이 짧아지게 되나 사용자는 일정 시간 이내에만 트랜잭션 확정되면 되므로 굳이 높은 수수료를 지불할 필요가 없고, 수수료에 따른 정확한 소요 시간도 알 수 없다. 따라서, 사용자의 입장에서는 트랜잭션 수수료를 최소화 하기 위해 트랜잭션 수수료에 따른 소요 시간을 예측할 수 있는 모델이 필요하다. 현재 비트코인 코어에서 예측 모델이 제공되고 있고 곳곳에서 보다 정확한 예측 모델에 대한 연구도 진행되고 있지만 비트코인 트랜잭션 수는 시시각각으로 급변하고 트랜잭션 수수료는 현재 Mempool 에 존재하는 트랜잭션의 수에 큰 영향을 받기 때문에 트랜잭션 수수료에 따른 소요 시간을 정확히 예측하기는 쉽지 않은 일이다.

본 논문에서는 이러한 문제를 해결하기 위하여 머신 러닝 기술을 도입 트랜잭션이 Mempool 에 들어올 당시 Mempool 에 존재하는 트랜잭션 수를 고려하여 트랜잭션의 트랜잭션 수수료와 트랜잭션이 확정되기까지의 시간과의 관계를 머신 러닝을 통해 학습시킴으로써 비트코인 트랜잭션 수수료 예측 모델을 만드는 것을 목표로 한다.

본 논문에서는 이후 비트코인에서 트랜잭션이 확정되기까지의 과정에 대해 다루며 현재 비트코인 코어에서 제공되는 트랜잭션 수수료 예측 모델에 대해 설명한다. 그 다음 본 논문에서 트랜잭션 수수료 예측 모델을 제작하기 위해 사용한 방법에 대해 소개하고 마지막으로 본 논문의 결론과 향후 과제에 대해 논의한다.

Ⅱ. 배 경

비트코인에서는 트랜잭션이 생성되면 Mempool 에 들어가게 되는데 트랜잭션이 확정되기 위해서는 트랜잭션이 블록에 포함되어야 한다. 채굴자는 작업 증명을 통해 Mempool 에 있는 트랜잭션을 블록에 포함시켜 블록을 생성하고 채굴 보상(Mining Reward)과 생성된 블록에 포함된 트랜잭션들의 트랜잭션 수수료를 받게 된다. 채굴자가 생성하는 블록의 최대 크기는 제한되어있기 때문에 채굴자가 최대의 이익을 얻기 위해서는 단순히 트랜잭션의 크기와 수수료의 비율 만을 비교해서는 안되나 일반적으로는 비율이 높은 트랜잭션을 우선적으로 블록에 포함시킨다. 따라서, 대부분의 예측 모델에는 트랜잭션 크기와 수수료의 비율을 사용하며 사용자는 이 비율을 바탕으로 자신의 트랜잭션의 크기에

따라 실제 트랜잭션 수수료를 계산한다.

비트코인 코어에서 제공하는 예측 모델의 경우 기존의 데이터들을 기반으로 사용자가 숫자 N 을 입력했을 때, 사용자의 트랜잭션이 N 블록 내에 포함되기 위해 지불해야하는 수수료의 최솟값을 예측한다. 이 모델에서는 최근에 생성된 블록들의 트랜잭션에서 트랜잭션 사이즈와 트랜잭션 수수료의 비율들을 참조하여 이를 특정한 구간에 따라 분류한 뒤, 각 구간에서 해당 트랜잭션들이 N 블록 안에 포함 되었는지를 조사한다. 이 때, 보다 최근에 생성된 블록일수록 정확한 예측이 가능하기 때문에 블록의 트랜잭션들을 분류할 때 최근에 생성된 블록에 더 높은 가중치를 둔다. 트랜잭션 사이즈와 트랜잭션 수수료의 비율이 높은 구간에서부터 시작하여 일정 비율 이상의 트랜잭션들이 N 블록 안에 포함되었다면 비율이 더 낮은 구간으로 이동하여 계산을 반복하고 더 이상 일정 비율 이상의 트랜잭션들이 N 블록 안에 포함되지 않았을 때 그 위의 구간에 해당하는 비율을 추천해주는 방식이다.

이 모델은 많은 사용자가 사용하고 최근에 생성된 많은 데이터를 기반으로 하고 있기 때문에 정확한 예측이 가능하다. 하지만 과거의 데이터만을 기반으로 하고 현재 Mempool 에 대한 정보가 없기 때문에 비트코인 트랜잭션 수에 급격한 변화가 나타났을 때 정확한 예측을 하지 못해 취약하다는 단점이 존재한다.

Ⅲ. 본 론

본 논문에서는 머신 러닝을 이용하여 트랜잭션 수수료와 트랜잭션이 확정되기까지의 시간을 학습시킴으로써 비트코인 트랜잭션 수수료 예측 모델을 만드는 것을 목표로 한다. 이를 수행하기 위하여 먼저 비트코인 트랜잭션들에 대한 데이터를 수집하여야 한다. 이를 위하여 비트코인 코어에서 제공되는 기능들을 이용하여 비트코인 트랜잭션들에 대한 데이터를 수집할 수 있는 프로그램을 개발하였다.

b2a302333cf1ff4698b5e7176fcac73d3b0230b654639667bddbd101e27b935d	77	1	55650
4be86a00a68fefe30eeb08e1431bc00855483e9d085e2bcc2c932fd90e49791a	67.29778	5	54117
2344656ebb9ad613dafab9d70e560ba8e09c0dba5506a19bd6dbaf133dbafa36	68.70229	4	55598
ca346208f72c22e6b41c843161b51f76c56354302de49d3ef0c65145a513d711	68.2973	1	55650
c8a8602bf364a0328333fc187ca084e6df95349bf2b13de780508e0a1b956a10	71.63393	2	55493
475fe13bcc007339830d9eac09f4e9a7bebaf86af01224accba94795d05851d9	79.69474	3	55372
d604d44211fb32415660d755a7ded198c83262330789dee44b4b3e7811b9082b	68.9148	2	55493
c017ba5e9c37165c7f28eb767c59c29139267e0ba7a6709bd9e03c05d7cd8685	73	3	55372
b1dc1283585c21c6db7a15502408821cf84736fe2885f236879d702e1dfbe23b	66.88789	1	55650
893d96fd133da13295df1d102fc16a01c5e1763f20ac8ca27fbfb4dac97df42d	70	4	55598
587d26e576cb041bb8d67b22d566b65c5dd6d6b3357a33258ab35fd75151ada0	69	3	55372
e045b02ece2a2ae35316b68aa4710a977cdf35e87b8fd12b4d4941b0aae1e8d4	72.96861	6	54060
fadb90d78a17486bf06e404615bf7fcd041423b6e7089f6667d0ae3fa7e4ea7e	69.09677	4	55598
c7c56b0c26e9e248365b302606932ae1794e8c20371e9fff45697be2051ba4fd	70.18767	5	54117
a792d214089499ca04f2e2a3b0590a431fc2128be8f76ae2313e32267d7fd764	74	1	55650
784fb11e11fd91618fd86e14531687841ebe9bd5303bb68ae66ffe3b1270ff60	73.98206	5	54117
eac006f636ab6c8237785a1b9354abcdd88bb6a6909585bc958d3fbb28f85139	63.18639	6	54060
123f7abbb3e8cab61684c419fc86e64322c315554bef7b642947c153236b3d1e	70.24324	2	55493
0abdb2ba4469ce005217265b6e7da295fd755cc8cd7da41435a0905c90a80900	72.32	3	55372
f4e51b8526f6889f16f8cf358186ec9a957d210448891d88da8735920f86a842	70	5	54117
84626262682e37bad348853187655cc030d0ea68fc6bb59251e2e07be7041cb1	68.60714	1	55650
92141ed7cc375640b516b0914d821d0187f6b6c69e87aa50e7db6c1340b06799	74.5445	6	54060

그림 1. 트랜잭션 데이터 예시

그림 1 은 수집한 트랜잭션 데이터의 예시이다. 첫 번째 열에는 트랜잭션의 ID 를, 두번째 열에는 트랜잭션 수수료를 트랜잭션의 크기로 나눈 값을 저장하였으며 이 단위는 sat/byte 이다. 세 번째 열은 트랜잭션이 Mempool 에 들어온 이후 확정되기까지 생성된 블록의 수를 의미하며 네 번째 열은 트랜잭션이 Mempool 에 들어왔을 때 Mempool 에 존재하는 트랜잭션 수를 의미한다.

다음으로는 머신 러닝 학습 모델을 구축해야 하는데, 이는 tensorflow 를 이용함으로써 구현하였다. 학습은 그림 1 과 같이 수집한 트랜잭션 데이터를 바탕으로 진행하며 수집한 데이터의 일부는 테스트 셋으로 사용하여 테스트 셋을 제외한 데이터로 학습을 진행한 후 테스트 셋을 이용하여 모델의 정확도를 테스트 한다. 트랜잭션 데이터 셋은 매우 크기 때문에 학습에는 미니 배치를 이용한 경사하강법을 활용하여 진행하였다.

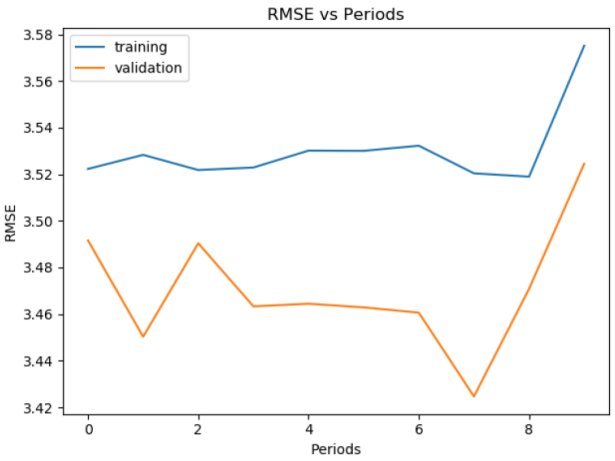


그림 2. 학습 결과 예시

이렇게 제작된 모델은 모델의 평균 제곱근 편차를 계산하여 정확도를 확인할 수 있다. 그림 2 는 학습한 모델에서의 평균 제곱근 편차와 이를 테스트 셋에 적용시킨 평균 제곱근 편차를 그래프로 나타낸 예시이다. 제작된 모델은 학습률과 배치 크기를 변화시킴으로써 정확도를 향상시키는 것이 가능하다.

Ⅳ. 결 론

본 논문에서는 머신 러닝을 이용한 비트코인 트랜잭션 수수료 예측 모델을 만들었다. 본 논문에서 제시한 모델의 정확도를 측정해본 결과 아직 유의미한 결과를 내지 못하여 성능의 개선을 필요로 한다. 따라서 이후 현 모델의 성능 개선을 수행할 예정이다. 또한, 현 모델은 linear regression 모델을 기반으로 제작되었기 때문에 추후 tree regression 모델과 deep neural network 모델 개발을 목표로 하고 있다.

본 논문에서 제시한 방법을 통해 만들어진 모델이 기존의 비트코인 거래 수수료 예측 모델보다 더 낮은 수수료로 정확한 예측이 가능해진다면 비트코인 유저들이 더 적은 수수료를 이용해 거래가 가능해질 것이다. 또한, 수수료의 저하는 비트코인 거래가 좀 더 활성화되는데 기여할 수 있을 것이다.

ACKNOWLEDGMENT

이 논문은 2019 년도 정부(과학기술정보통신부)

의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구 임 (No.2018-0-00539)

참 고 문 헌

[1] Al-Shehabi, Abdullah. "Bitcoin Transaction Fee Estimation Using Mempool State and Linear Perceptron Machine Learning Algorithm." (2018).

한국통신학회 통신망운용관리연구회
2019년 통신망운용관리 학술대회 논문집
Proceedings of KNOM Conference 2019

ISSN : 2586-0232 (Online)
2019년 5월 31일 인쇄
2019년 5월 31일 발행

발행인/석승준 운영위원장
편집인/김경백 출판위원
발행처/한국통신학회 통신망운용관리연구회
서울시 서초구 서초동 1330-8 현대기림오피스텔 1504동 6호
전 화 : 02-3453-5555
홈페이지 : www.knom.or.kr
디자인 및 편집/ 전남대학교 DNSLAB