논문 22-47-03-06 The Journal of Korean Institute of Communications and Information Sciences '22-03 Vol.47 No.03
https://doi.org/10.7840/kics.2022.47.3.453

이더리움 네트워크에서 액티브 프루빙을 이용한 정확도 높은 토폴로지 탐색 방법

맹 수 훈◆, 주 홍 택°

A High-Accuracy Dynamic Topology Inference Method Using Active Probing in the Ethereum Network

Soohoon Maeng◆, Hongtaek Ju°

요 약

블록체인 네트워크에서 DDoS 문제, 51% 공격, 시빌 공격과 같은 보안 문제와 중복 메시지, 영향력 있는 피어 선정을 통해 네트워크의 성능을 높이기 위한 연구는 네트워크에서 동적으로 변화하는 노드에 대한 분석이 기초가 되어야한다. 비트코인 네트워크에서는 이중 지불 트랜잭션, 어드레스프로브(AddressProbe), 노드프로브(NodeProve)를 이용하여 토폴로지를 분석하고 있지만 이더리움 네트워크에서 동적으로 변화하는 토폴로지에 대한 연구는 수행되지 않았다. 본 논문에서는 패시브 프루빙 방법을 활용하여 특정시간 네트워크 참여 중인 노드를 탐색하는 액티브 프루빙 방법을 제시한다. 패시브 프루빙을 통해 발견된 131,776개의 노드와 8,643,225의 연결성을 이용하여 특정시간 동안 이더리움 네트워크에 참여한 13,261개의 노드를 확인했다. 동적 토폴로지 분석을 위해 평균 연결 차수, 평균 연결 경로 수, 직경, 결집 계수, 매개 중심 등을 분석하고 토폴로지를 시각화 했다. 본 논문의 결과는 이더리움 네트워크에서 트랜잭션, 블록의 전파 경로를 추적하는 연구에 적용할 수 있다. 또한 영향력 있는 피어 선정을 통해 네트워크 성능향상, 중복 메시지 제거에 대한 연구의 기초 자료로 활용될 수 있다.

키워드 : 블록체인, 이더리움, P2P 네트워크, 토폴로지, 노드 탐색, 액티브 프루빙
Key Words : Blockchain, Ethereum, P2P Network, Topology, Node Discovery, Active Proving

ABSTRACT

Security issues such as DDoS problems, 51% attacks, and Civil attacks in blockchain networks, duplicate messages, and research to increase network performance through influential peer selection should be based on analysis of dynamically changing nodes in the network. Bitcoin networks analyze topologies using Double Spending transactions, AddressProbe, and NodeProve, but no research has been conducted on dynamically changing topologies in the Ethereum network. In this paper, an active probing method for searching for nodes participating in a network at a specific time using the passive probing method is presented. We identified 131,776 nodes found through passive probing and 13,261 nodes participating in the Ethereum network for a specific time using the connectivity of 8,643,225. For dynamic topology analysis, the average degree, average path length, diameter, average clustering coefficient, and Betweenness centrality were analyzed and the topology was visualized. The results of this paper can be applied to studies that track the propagation path of

※ 본 연구는 한국연구재단 기초연구사업(NRF-2018R1D1A1B07050380) 지원 및 계명대학교 네트워크 연구실 관리로 수행되었습니다.
◆ First Author : Daegu-Gyeongbuk Medical Innovation Foundation, shmaeng@kmedihub.re.kr, 정회원
° Corresponding Author : Keimyung University Department of Computer Engineering, juht@kmu.ac.kr, 종신회원
논문번호 : 202111-305-B-RN, Received November 5, 2021; Revised December 15, 2021; Accepted January 19, 2022

transactions and blocks in the Ethereum network. In addition, it can be used as basic data for research on improving network performance and removing duplicate messages through influential peer selection.

Ⅰ. 서 론

이더리움(Ethereum)은 비탈릭 부테릭(Vitalik Buterin)에 의해 튜링 완전(Turing Completeness)언어를 접목시켜 여러 가지 탈중앙화 어플리케이션(Decentralized Application)구현이 가능한 블록체인 플랫폼이다[1]. 이더리움 네트워크는 P2P 네트워크 중에 잘 알려진 카뎀리아(Kademlia)[2]를 사용하여 P2P 네트워크에 참여하고 새로운 노드를 탐색하며 이웃 노드와의 연결을 변경한다.

이더리움 네트워크는 DDoS 공격, 51% 공격, 시빌 공격과 같은 보안 공격[3,4]에 대처해야 하고 네트워크가 확장됨에 따라 초당 거래 처리 속도(TPS)가 느려지는 확장성(Scalability)문제를 해결해야 한다. 이러한 문제 해결 방안은 동적으로 변화하는 토폴로지 분석과 토폴로지를 구성하고 있는 노드 간 연결에 대한 기초 연구를 바탕으로 도출되어야 한다. 이더리움 네트워크에서 노드 탐색을 기반으로 토폴로지를 분석하기 위한 연구[5]와 피어의 속성을 분석하는 연구[6]가 있었으나 연결 정보를 패시브 프루빙(Passive Probing)으로 수집한 후에 IP를 기반으로 피어 간 연결을 기반으로 액티브 프루빙(Active Probing)과정을 수행하여 이더리움 네트워크의 토폴로지 분석에 대한 정확도를 높인 연구는 없었다.

본 논문에서는 이더리움 네트워크에 참여하고 있는 노드를 카뎀리아 방식을 활용하여 탐색하고 동적으로 변화하는 토폴로지를 확인하는 방법을 제안한다. 본 논문에서는 이더리움 네트워크의 토폴로지를 측정하기 위해 카뎀리아를 기반으로 노드를 탐색하고 패시브 프루빙 데이터를 수집한다. 이더리움 카뎀리아를 활용하여 탐색한 노드는 실제로 동작하지 않는 이웃 피어의 정보가 포함될 수 있다. 이는 블록체인 네트워크의 천(Chrun)현상[7]과 사용자들의 네트워크 참여 의사에 따라 변경될 수 있다.

수집된 패시브 프루빙 데이터는 액티브 프루빙 과정을 노드를 통해 액티브 프루빙 과정을 수행한다. 액티브 프루빙 토폴로지는 최소한의 시간 동안 변경되지 않은 상태로 유지된다고 가정한다. 이를 통해 이더리움 네트워크에서 노드 활성화 유무를 통해 파악된 토폴로지는 그래프 이론에 입각하여 특징을 분석하고 이더리움 네트워크의 토폴로지 측정에 대한 정확도를 높인다. 본 논문의 결과는 향후 블록체인 토폴로지에서 영향력 있는 피어 선정을 통한 네트워크 성능 향상, 중복 메시지 제거에 대한
기초 연구로 활용될 수 있다.

본 논문의 구성으로 2장은 기존 비트코인 토폴로지 분석 연구와 기존의 이더리움 토폴로지 분석 결과를 확인한다. 또한 이더리움 네트워크에서 기본적인 노드 탐색 방법에 대해 논한다. 3장은 이더리움 네트워크에서 동적 토폴로지 탐색 방법과 분석 방법 대해 설명한다. 4장에서는 실험에 대한 분석 결과를 제시하고 5장은 본 논문의 결론을 도출한다.

Ⅱ. 관련 연구

2.1 비트코인 토폴로지 분석

M. Grundmann 등[8]은 비트코인 네트워크에 참여하고 있는 노드에서 이웃 노드로 트랙잭션을 생성해서 보내고 수신함으로써 탐색하는 방법을 제안했다. 특정 노드의 이웃 노드를 확인하는 방법으로 각 노드마다 다른 트랜잭션을 보내거나 이중 지불 트랜잭션을 보내는 방법으로 탐색했고 거래 수수료문제로 인해 메인넷에 적용은 어렵다는 결론을 도출 했다.

Andrew Miller 등[9]은 어드레스 프로브(AddressProve)을 통해 비트코인 네트워크에 참여한 노드들 간의 링크를 발견하고 동적 토폴로지에 적용하는 방법을 제안했다. 그 결과로 비트코인 네트워크에서 노드 간 서로 연결된 커뮤니티가 존재를 확인하고 측정된 동적 토폴로지의 결과를 도출했다.

Essaid Meryam 등[10]은 비트코인 네트워크에서 활동하고 있는 노드를 재귀적으로 스캔하여 노트를 탐색하는 노드프로브(NodeProbe)를 도입하는 방법을 제시했다. 이를 통해 비트코인의 시간 변화에 따라 활동 노드 수, 영구 노드 등의 변화를 확인하고 노드 수가 같은 랜덤 네트워크 모델보다 커뮤니티 수가 4배 이상 많고 특정 노드가 비트코인 네트워크의 백본 역할을 수행하고 있다는 결과를 확인했다.

비트코인 네트워크의 토폴로지 분석은 이중 지불 트랜잭션, 어드레스 프로브(AddressProve), 노드프로브(NodeProbe)등을 이용하여 동적 토폴로지에 대한 활발한 연구가 수행되고 있다.

2.2 이더리움 토폴로지 분석

Gao, Yue 등[13]은 이더리움 네트워크에 참여한 노드ID를 수집하기 위해 RLPx단계의 노드 탐색 횟수를 늘렸다. 이를 통해 2개월 동안 60개의 노드에서 하루 평균 508,467개의 노드ID를 수집했고 동일한 IP주소에서 2개 이상의 노드 ID를 사용하는 IP주소가 발견되었다. 이는 이더리움 네트워크에 연결되어 있는 노드 중 NAT뒤에서 참여하고 있다는 사실을 확인했고, 이 연구를 통해 52,554개의 IP주소를 탐색했다. 하지만 특정 시간 이더리움 네트워크에 참여하고 있는 노드들을 더 많이 확인하는 연구보다 NAT뒤에 활동하는 노드들에 대한 연구를 중심으로 진행했다.

Seoung Kyun, Kim 등[5]은 NodeFinder를 구현하여 이더리움 네트워크에 참여하고 있는 클라이언트의 종류, 상태, 특징 등을 주기적으로 확인했다. 이를 통해 이더리움 네트워크에서 동작하는 클라이언트에 대한 분석을 진행했다. 또한 하루 동안 수집된 클라이언트의 노드 ID를 바탕으로 15,454개의 노드를 탐색했다. 이 수치는 기존 이더리움 노드 정보 제공 웹[16]보다 약 3배 많은 노드를 발견했지만 노드 간 연결 관계를 확인할 수 없기 때문에 토폴로지 분석 및 시각화에 대한 연구는 수행하지 못했다.

이전 연구[11,12]에서 기본 동작으로 탐지 노드와 연결 되어 있는 노드ID를 수집하여 로그에 저장했다. 기존 연결 방법 보다 더 많은 피어를 탐색하기 위해서 그림 1과 같이 30초 간격으로 버켓(Bucket)의 피어를 삭제하는 과정을 추가하는 패시브 프루빙 방법을 고안했다. 이때 피어 삭제 과정은 탐지 노드에 영향을 미치지 않기 위해 카뎀리아(Kademlia)[2]의 노드 탐색 매커니즘을 참고하고 30초마다 수행한다. 정적 토폴로지 탐색 실험 결과 하루 동안 약 2만개의 노드를 발견했다. 또한 이더리움 네트워크에 참여하고 있는 탐지 노드에 부하가 가지 않기 위해 이웃 피어와 연결하는 이더리움 클라이언트의 기본 로직(Logic)을 사용했다.

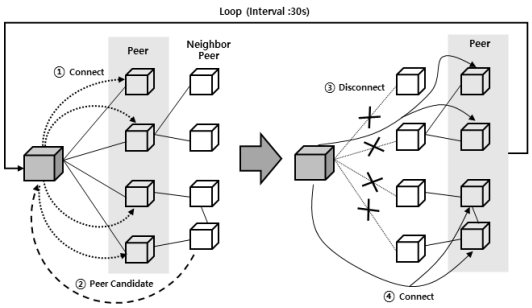

그림 1. 피어 연결 삭제 및 이웃 피어 연결 과정 예시
Fig. 1. Example of peer connection deletion and neighbor peer connection process

하지만 패시브 프루빙과정은 단순히 노드 탐색 개념으로 이더리움 네트워크에 참여하고 있는 모든 노드들을 탐색할 수 없다는 한계가 존재한다. 따라서 특정 시간 이더리움 네트워크의 토폴로지를 확인할 수 없었다.

이처럼 특정 시간 이더리움 네트워크의 참여 유무와 노드 간 연결 정보 확인을 통해 이더리움 네트워크에서 정확도 높은 토폴로지에 대한 분석 및 시각화 연구는 수행되지 않았다.

2.3 이더리움 네트워크 노드 탐색 과정

이더리움 네트워크는 탈중앙화 분산 P2P 네트워크를 구성하기 위해 devp2p[18] 프로토콜을 사용하여 이웃 노드를 탐색한다. devp2p는 카뎀리아[2]를 기반으로 구현되었고 분산해시테이블(Distributed Hash Table)을 사용한다. 카뎀리아는 UDP를 사용하여 오버레이 네트워크를 구축하고 참여 노드는 공개키로 구현된 ID로 식별한다. 노드 간의 논리적 XOR거리를 측정하여 노드를 탐색하고 이웃 노드의 정보는 16개의 버켓(Bucket)에 수집한다. 현재 이더리움 버전 4에서 사용 중인 카뎀리아 노드 탐색 프로토콜은 4가지이며 표 1과 같다.

이더리움 클라이언트는 피어 목록이 비어 있는 경우 클라이언트에 명시된 부트스트랩 노드(Bootstrap_Ndoe)의 버켓에 저장된 피어 목록을 요청하고 부트스트랩 노드의 이웃 피어를 응답받는다. 버켓에 저장된 피어들의 목록은 PING 메시지를 통해 주기적으로 활동 유무를 확인하며 피어 목록을 갱신한다. 이후 이더리움 클라이언트는 내부에 저장된 피어 목록을 기반으로 이더리움 네트워크의 노드 탐색을 통해 전달 받은 이웃 노드의 정보를 저장하는 과정

표 1. 이더리움 네트워크 노드 탐색 프로토콜
Table 1. Ethereum network node discovery protocol

Protocol	Description
PING	버켓에 저장된 노드의 활동 유무를 확인
STORE	탐색된 노드의 정보를 키, 값을 쌍으로 저장
FIND_NODE	노드에 가장 가까운 이웃 피어에 대한 ID를 요청
FIND_VALUE	노드에 가장 가까운 이웃 피어에 저장 값을 요청

을 반복적으로 수행한다. 이웃 피어는 이더리움 네트워크에 참여하고 있는 피어와 연결된 노드이며 이웃 피어는 IP, Port로 응답받으며 FIND_NODE 메시지로 피어로 부터 전달 받는다.

Ⅲ. 동적 토폴로지 탐색 방법 및 분석 방법

3.1 동적 토폴로지 탐색 방법

이더리움 네트워크의 동적으로 변화하는 토폴로지를 탐색하기 위해서는 탐지노드, 패시브 프루빙, 액티브 프루빙과 같은 3가지 방법을 수행한다. 이전 연구[11,12]에서 고안한 패시브 프루빙은 탐지 노드와 전처리 기능(PreProcess)이 포함되고 본 논문에서 고안한 동적으로 변화하는 이더리움 네트워크를 분석하기 위한 액티브 프루빙방법은 분석 엔진(Analysis Engine) 과정과 같이 동작하고 전체 시스템은 그림 2와 같이

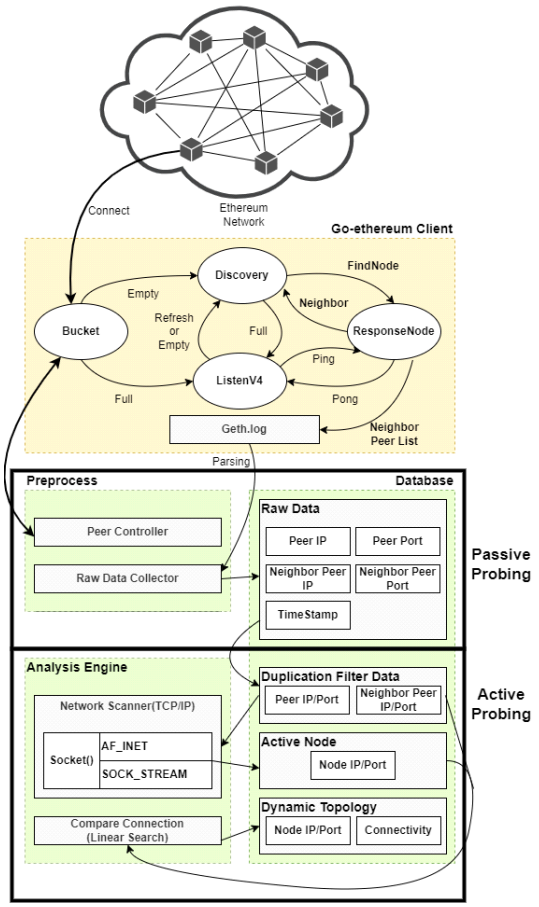

그림 2. 동적 토폴로지 탐색 및 분석 시스템 구성
Fig. 2. Dynamic topology exploration and analysis system configuration

구성된다.

탐지 노드는 FIND_NODE을 통해 응답 받은 이웃 피어 정보를 수집하기 위해 이더리움 클라이언트의 로그파일에 저장하는 기능을 추가하고 수집된 정보는 피어와 이웃 피어의 연결성을 함께 로그 파일(Geth.log)에 저장한다. 또한 Data Collector과정에서 로그파일 정보를 파싱(Parsing)하여 피어 IP, 피어 Port, 이웃 피어 IP, 이웃피어 Port, 타임 스탬프정보를 Raw Data 데이터베이스에 저장한다.

패시브 프루빙과정은 그림 2의 Passive Probing 부분에 해당하고 전처리(PreProcess)에서 구현된다. 전처리과정은 Peer Controller, Raw Data Collector로 나누어지고 Peer Controller과정은 많은 피어를 수집하기 위해 탐지 노드(Go-ethereum Client)와 연결된 피어를 버켓에서 삭제하는 과정을 30초 간격으로 반복 수행한다. 이후 Data Collector를 통해 노드 정보를 로그 파일에 수집한다.

본 논문에서 제안한 액티브 프루빙과정은 분석엔진(Analysis Engine)을 통해 구현하고 그림 2의 Active Probing방법에 해당한다. 분석엔진은 Network Scanner, 연결 비교(Compare Connection)과정을 수행하며 이더리움 네트워크에 참여하고 있는 노드를 탐색한다.

분석엔진을 통해 액티브 프루빙과정을 수행하기 앞서 Duplication Filter과정을 통해 수집된 모든 데이터를 표준화하는 과정을 수행한다.

Duplication Filter과정은 Raw Data에 저장된 피어의 IP, Port를 중복 제거하고 피어와 이웃 피어로 구성된 연결 정보에 대한 중복된 데이터를 제거한 후 Duplication Filter Data 데이터베이스에 저장한다. Duplication Filter Data 데이터베이스는 피어와 이웃 피어의 IP, Port의 중복 값을 제거하고 피어 IP와 Port 값이 쌍으로 저장되며 이 과정에서 피어와 연결되어 있는 이웃 피어의 연결성(Connectivity)은 배제되어도 된다.

Network Scanner과정은 Duplication Filter 데이터베이스에 저장된 노드의 IP, Port를 대상으로 TCP/IP Ping을 통해 이더리움 네트워크에서 해당 노드의 동작 유무를 조사하고 그 결과를 Active Node 데이터베이스에 저장한다. 이 과정이 기존 연구에 추가된 액티브 프루빙 과정이다. Network Scanner는 TCP/IP Ping 패킷을 보내며 실험 속도를 높이기 위해 50ms 간격의 쓰레드(Thread)로 구현되었다. 활동 중인 노드에 대한 신뢰성을 높이기 위해 2회 수행한다. 신뢰성을 높이기 위해서 더 많은 횟수로 수행할 수 있지만

실험 시간에 대한 제약 때문에 2회 시행한다. 또한 이더리움 네트워크에서 카뎀리아의 노드 탐색과정에서 'Hello'메시지는 500ms의 대기시간을 가지므로 Network Scanner 모듈은 응답 대기 최대 시간은 1초로 설정해 충분한 응답 대기 시간을 가지도록 설계했다[13]. Network Scanner는 파이썬 기반으로 Network Socket에서 생성 함수 AF_INET, SOCK_STREAM를 사용하여 신뢰성을 높였다.

로그 파일로 수집된 패시브 프루빙 정보와 Ping 프로토콜로 확인한 액티브 프루빙 정보의 정확도 향상 정도를 분석하기 위해 Raw Data 데이터베이스와 Active Node 데이터베이스의 결과를 서로 비교하는 분석은 Compare Connection 모듈이 수행한다. 동적 토폴로지는 Network Scanner 모듈의 결과로 활동 중인 노드의 IP, Port와 Raw Data의 피어와 이웃 피어 연결을 비교하여 동적으로 변화하는 이더리움 네트워크의 토폴로지를 확인한다. Dynamic Topology 데이터베이스에서 Network Scanner와 Raw Data을 비교 알고리즘은 단순하게 선형 탐색(Linear Search) 방법을 사용하여 각 요소를 순차적으로 확인한다.

3.2 동적 토폴로지 분석 방법

본 논문에서는 동적 토폴로지를 분석하기 위해 평균 연결 차수, 평균 경로 길이, 직경, 평균 클러스터링 계수, 근접 중심성, 매개 중심성으로 나누어 표 2와 같은 수식을 이용하여 분석을 진행했다.

표 2. 동적 토폴로지 분석 내용
Table 2. Dynamic topology analysis

Division	Equation
평균 연결 차수 (Average Degree)	$D_{Avg}\frac{1}{N}\sum_{1}^{N}K_i$
평균 연결 경로 길이 (Average Path Length)	$APL_{Avg}=\frac{\sum_{i,j}^{N}Dis(n_i,n_j)}{\sum_{i,j}^{N}Path(n_i,n_j)}$
직경(Diameter)	$D=\max_{i,j}(Path(i,j))$
평균 결집 계수 (Average Clustering Coefficient)	(1) $C_i=k_i\frac{2T_i}{(k_i-1)}$ (2) $C_{Avg}=\frac{1}{N}\sum_{N}^{i}C_i$
근접 중심성 (Closeness Centrality($i\neq j$))	$C_c(N_i)=\frac{1}{[\sum_{j=1}^{g}=d(N_i,N_j)]},i\neq j$
매개 중심성 (Betweenness Centrality)	$C_B(N_i)=\sum_{j<k}\frac{g_{jk}(N_i)}{g_{jk}}$

평균 연결 차수(Average Degree)는 네트워크에 참여하고 있는 노드들에 대한 연결 수(K_i)를 노드의 수(N)로 나누어 전체 노드의 평균 연결 차수를 계산한다. 임의의 노드와 이웃하는 노드들 사이에 연결된 선들의 평균을 의미하며 평균 경로 길이는 토폴로지 내 임의의 두 노드가 연결되어 있을 때 반복하지 않는 선들의 연결 평균 길이를 의미한다.

평균 연결 경로 길이(Average Path Length)는 임의의 노드 i에서 노드j까지의 연결 수의 평균을 의미한다. 네트워크에서 평균 경로 길이(APL_{Avg})를 측정하기 위해 네트워크의 모든 경로 길이($Dis(n_i,n_j)$)합과 네트워크 내 모든 경로 수($Path(n_i,n_j)$)를 각각 계산한 다음 수식 (2)와 같이 모든 경로 길이의 합($\sum_{i,j}^{N}Dis(n_i,n_j)$)을 경로 수의 합($\sum_{i,j}^{N}Path(n_i,n_j)$)으로 나누어 평균 경로 길이($APL_{Avg}$)를 계산한다.

직경(Diameter)은 임의의 노드 i에서 노드j까지의 최대 연결 수를 의미한다. 토폴로지 내의 최대 거리를 의미하며 직경이 클수록 네트워크에서 블록과 트랜잭션이 전달되어야할 경로가 증하기 때문에 블록체인 네트워크의 전파, 전달 지연과 관련이 있다.

평균 결집 계수(Average Clustering Coefficient)는 특정 노드와 연결된 임의의 두 노드가 서로 연결 되어 있을 확률을 의미하며 임의의 네트워크에서 군집이 존재하지 않으면 평균 결집 계수 값은 0에 가깝다. 이 결집 계수는 각각의 노드들의 커뮤니티 형성에 대한 경향을 확인할 수 있다. 각각의 노드에 대하여 클러스터링 계수를 계산한 것이 지역 결집 계수(Local Clustering Coefficient)이며 수식(1)와 같이 계산된다. 평균 결집 계수(Average Clustering Coefficient)는 모든 지역 결집 계수의 평균이고 수식(2)와 같이 나타낸다. 수식에서 N은 총 노드의 수이다. T_i는 노드i를 연결하는 삼각형의 수이고 K_i는 노드i의 연결 차수를 나타낸다. 따라서 C_{Avg}는 모든 로컬 결집 계수 C_i의 평균을 나타내고 이더리움 네트워크에서 임의의 두 노드가 연결될 확률을 측정하는 지표가 된다.

근접 중심성(Closeness Centrality)은 전체 네트워크에서 임의의 노드가 중앙에 위치하고 있는지를 확인한다. 또한 근접 중심성은 중요한 노드일수록 다른 노드까지 도달하는 경로가 짧은 것을 의미하며 작은 값일수록 중심에 가깝다. 이는 네트워크에 참여하고 있는 임의의 모든 두 노드 사이의 최단 경로를 계산하고 다른 모든 노드에 대한 최단 거리의 합으로 계산한다.

매개 중심성(Between Centrality)은 임의의 두 노드

사이의 가장 짧은 경로가 노드를 거쳐가는 노드 수를 의미하고 매개 중심성이 높은 노드는 다른 노드들 사이에서 중간 역할을 하며 네트워크에 영향력을 확인하는 지표다. 두 노드j, k사이에 존재하는 임의의 노드i를 경유하는 횟수를 수치화한 값이다. 이때 g_{jk}는 두 노드j, k사이에 존재하는 임의의 노드i의 최단 거리 경로 수를 나타낸다.

Ⅳ. 동적 토폴로지 분석 결과

4.1 실험 환경

본 연구에 사용된 탐지 노드는 많이 사용이 되는 이더리움 클라이언트(Ethereum Client)[14]인 Go-ethereum을 기반으로 구현했다. 탐지 노드가 설치된 서버는 Intel Core i7-6700 CPU@3.40GHZ 8코어, 1TB SSD, 16GB의 RAM을 사용하며 운영체제는 Linux 18.04 버전이며 이더리움 노드의 최소 요구사항[15]을 충족했다. 이더리움 네트워크 노드 탐색을 위한 클라이언트는 Go-ethereum은 1.9.12 버전을 사용하고 Go-Language는 1.13.6을 설치했다. 탐지 노드는 일반 인터넷 서비스 제공자(ISP)에 연결되고 접속 속도는 약 20Mbps이며 파이어월과 같은 보안 장비가 없이 인터넷에 연결되어 있다. 패시브 프루빙 데이터 수집 기간은 2020년 12월부터 2021년 2월까지 2개월이고 액티브 프루빙과정을 통한 토폴로지 조사는 2021년 2월 28일 12시에 약 30분 동안 수행했다. 이더리움 네트워크는 노드들의 참여와 탈퇴를 반복하고 새로운 노드가 추가되기 때문에 패시브 프루빙 데이터는 수집 기간이 필요하다. 패시브 프루빙 데이터 수집 기간은 실험 환경으로 인해 2개월로 설정되었지만 기간을 확장하고 기존 1개의 탐지 노드보다 더 많은 탐지노드를 구성하면 노드 간 연결 정보를 추가적으로 수집할 수 있기 때문에 정확도 높은 토폴로지를 측정할 수 있다.

4.2 이더리움 네트워크 노드 탐색 결과

본 연구는 이더리움 네트워크에서 참여하는 클라이언트 전체 로직에 영향을 주지 않는 패시브 프루빙 데이터를 저장했고 2개월간 수집된 패시브 프루빙 데이터는 131,776개의 피어와 각 피어에 연결된 8,643,225개의 이웃 피어에 대한 연결을 확인했다. Duplication Filter의 과정에서 중복을 제거하여 총 131,776개의 노드 IP를 확인했고 수집 기간 동안 전체 이더리움 네트워크에 한번이라도 참여한 노드들의 수이다. 또한 피어와 이웃 피어의 연결도 중복을 제거하여 총 1,464,853개의 연결이 수집되었다.

Network Scanner과정은 Duplication Filter과정에서 수집된 데이터를 이용하여 약 30분 동안 수행했고 최종적으로 13,261개의 노드가 동작 중인 것이 확인되었다. 이 결과는 상용화된 이더리움 노드 추적 웹에서 발견된 노드의 1.5배 이상이다. 또한 동작 중인 피어와 동작 중인 이웃 피어의 연결을 Network Scanner와 Raw Data를 선형 탐색한 결과 총 353,606개의 연결을 확인했다. 이더리움 네트워크에 참여하고 있는 노드를 탐색에 관한 대표 연구 결과는 표 3과 같다. NodeFinder[5]를 제외한 이더리움 노드 정보 제공 웹[16,17]과 Gencer[13] 보다 최대 약 3배 더 많은 노드를 발견했다. 하지만 노드 정보 제공 웹과 Gencer는 하루 동안의 노드를 정적으로 탐색하는 방법으로 구현했다. 또한 NodeFinder는 단순히 가장 좋은 성능을 나타내는 것처럼 보이지만 NodeFinder는 동적으로 변화하는 이더리움 네트워크를 분석할 수 없다. 하지만 본 연구에서 고안한 액티브 프루빙 방법은 30분 동안 13,261개의 노드를 검증했고 동적 토폴로지 분석에 효과적이다.

표 3. 이더리움 네트워크 노드 탐색 모델 별 노드 수 비교
Table 3. Comparison of the number of nodes by ethereum network node discovery model

Discovery Model	Date	Node Count	Point
탐지 노드	2021/02/28, (12:00pm ~ 12:30pm)	13,261	동적
Ethernodes[16]	2021/02/28	4,381	정적
Etherscan[17]	2021/02/28	6,496	정적
NodeFinder[5]	2018/04/23	15,454	정적
Gencer et al.[13]	2018/3/1 ~ 2018/4/11	4,302	정적

4.3 이더리움 동적 토폴로지 분석 결과

실험 결과 특정 시간 이더리움 네트워크에 참여 중인 13,261개의 노드들은 총 353,606개의 연결이 존재했고 분석된 결과는 표 4와 같다. 노드들의 평균 차수는 27.583이고 이는 피어가 이웃 피어의 평균 연결 수를 의미한다. 평균 경로 길이는 각각의 노드가 제 2의 노드에게 도달하는 최소한의 경로의 평균으로 3.694이고 이는 임의의 두 노드 간 평균거리를 의미한다. 이더리움 네트워크의 직경은 8로 관측되었으며 노드 간 밀도를 나타내는 결집계수는 0.478의 수치를 보여준다. 근접 중심도(Closeness Centrality)와 매개 중심

표 4. 동적 토폴로지 분석 결과
Table 4. The result of Dynamic topology analysis

Division	Result
Average Degree	27.583
Average Path Length	3.694
Diameter	8
Average Clustering Coefficient	0.478
Closeness Centrality	0.19
Betweenness Centrality	0.001

도(Betweenness Centrality)의 평균은 표 4와 같고 노드의 연결 차수가 높은 노드가 매개 중심도와 근접 중심도 모두 높은 수치로 나오지 않았다. 이는 연결 차수가 높은 노드가 토폴로지의 중심에 있지 않다는 결과를 의미한다.

이더리움 네트워크에서 활동 중인 노드를 단순히 연결 차수에 따라 헤비노드(50개 이상 연결), 미들 노드(2~50개 연결), 라이트 노드(1개 연결)로 분류하여 4,860개의 헤비노드, 7,613개의 미들 노드, 788개의 라이트 노드를 발견했다. 또한 근접 중심성과 매개 중심성은 35.xx.xx.xx가 0.484, 0.087로 가장 높은 결과를 보였고 특정 노드의 정보보호를 위해 IP주소는 8비트만 공개한다.

본 연구를 통해 연결 차수가 가장 높은(1,146개)노드는 부트스트랩 노드가 아닌 아마존 ISP를 사용하고 있다. 또한 연결 차수가 높은 상위 10개의 IP를 부트스트랩 노드와 비교해본 결과 상위 10개의 노드는 일반적으로 이더리움 네트워크에 참여하며 이더리움 네트워크에 영향력 있다는 사실을 확인했다.

Compare Connection단계의 결과인 353,606개의 연결성을 바탕으로 이더리움 네트워크의 토폴로지를 그림 3과 같이 시각화했다.

본 논문에서는 이더리움 네트워크에 참여하고 있는 노드들의 커뮤니티를 시각화하여 직관적으로 확인할 수 있다. 실험 결과 총 9개의 커뮤니티를 발견했고 연결 차수가 가장 높은 노드를 중심으로 그림 3의 중심에 형성된 커뮤니티를 확인했다. 또한 중심에 형성된 커뮤니티의 가장자리에 8개의 커뮤니티가 형성되어 이더리움 네트워크에 참여하고 있는 사실을 확인했다. 또한 연결차수가 높은 헤비노드는 커뮤니티의 중심에 많이 포진하고 가장자리 8개의 커뮤니티는 헤비, 미들 노드를 중심으로 라이트 노드들이 연결되어 이더리움 네트워크를 구성하고 있는 사실을 발견했다. 이는 네트워크에 영향력을 작게 미치는 노드 즉 네트워크에

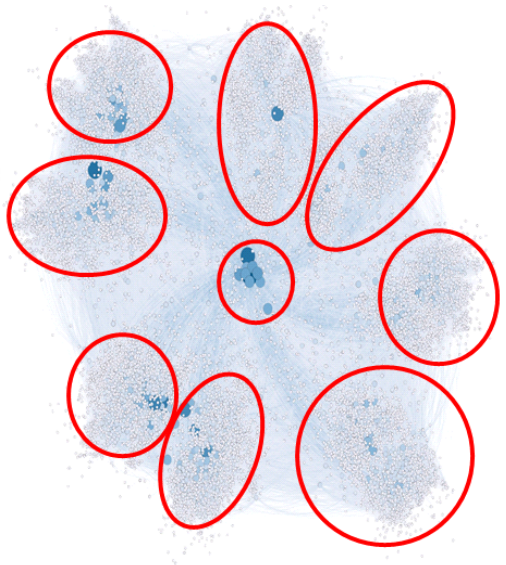

그림 3. 이더리움 동적 토폴로지 스냅샷
Fig. 3. The snapshot of Ethereum Dynamic topology

참여와 탈퇴를 반복하는 노드들을 제외한 결과이고 정확도 높은 토폴로지를 확인할 수 있다.

V. 결 론

본 논문에서는 이더리움 네트워크에서 동적으로 변화하는 토폴로지를 액티브 프루빙 방법을 사용하여 정확도 높은 토폴로지 탐색 방법을 제시한다.

이전 연구[11,12]의 노드 탐색과정을 반복하여 패시브 프루빙 데이터를 수집한다. 패시브 프루빙 데이터는 탐지 노드와 연결된 피어의 노드 목록이고 이더리움 네트워크에 특정 시간동안 참여 유무를 확인하기 어렵다. 이에 따라 본 논문에서는 액티브 프루빙 방법을 사용하여 이더리움 네트워크에 참여하고 있는 노드의 활성화 유무를 확인한다. 이더리움 네트워크의 토폴로지를 측정하기 위해서는 패시브 프루빙 데이터와 액티브 프루빙 데이터를 선형 탐색(Linear Search)방법으로 토폴로지 데이터를 수집 한다. 이더리움 동적 토폴로지 실험 결과 13,261개의 노드와 353,606개의 노드들의 연결을 확인했다. 이는 이더리움 노드 추적 웹에서 발견된 노드의 1.5배 이상 이더리움 네트워크에 참여하고 있는 노드를 발견했다.

본 연구는 이더리움 네트워크에서 동적으로 변화하는 토폴로지 분석은 이더리움 네트워크에서 노드의 반복적 네트워크 참여와 이탈한다는 사실을 확인할 수 있다. 또한 이더리움 토폴로지의 분석 결과와 시각

화는 이더리움 네트워크를 이해하는데 사용된다. 따라서 향후 블록체인 토폴로지에서 영향력 있는 피어 선정을 통한 네트워크 성능 향상, 중복 메시지 제거에 대한 연구를 진행할 예정이다.

References

[1] V. Buterin, "*A next-generation smart contract and decentralized application platform*," Ethereum White Paper 3.37, 2014.

[2] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," *Int. Wkshps. Peer-to-Peer Syst.*, Springer, Berlin, Heidelberg, 2002.

[3] M. Saad, et al., "Exploring the attack surface of blockchain: A systematic overview," arXiv preprint arXiv:1904.03487, 2019.

[4] X. Wang, et al., "Attack and defence of ethereum remote apis," *2018 IEEE GC Wkshps*, Abu Dhabi, United Arab Emirates, 2018.

[5] S. K. Kim, Z. Ma, S. Murali, J. Mason, A. Miller, and M. Bailey, "Measuring ethereum network peers," in *Proc. Internet Meas. Conf. 2018*, pp. 91-104, Oct. 2018.

[6] Y. Gao, et al., "Topology Measurement and Analysis on Ethereum P2P Network," *2019 ISCC*, Barcelona, Spain, 2019.

[7] M. A. Imtiaz, et al., "Churn in the bitcoin network: Characterization and impact," *2019 IEEE ICBC*, Seoul, Korea, May 2019.

[8] M. Grundmann, T. Neudecker, and H. Hartenstein, "Exploiting transaction accumulation and double spends for topology inference in bitcoin," *Int. Conf. Financial Cryptography and Data Secur.*, Springer, Berlin, Heidelberg, 2018.

[9] A. Miller, et al., "Discovering bitcoin's public topology and influential nodes," *Comput. Sci.*, 2015.

[10] M. Essaid, S. Park, and H. Ju, "Visualising bitcoin's dynamic P2P network topoogy and performance," *2019 IEEE ICBC*, Seoul, Korea, May 2019.

[11] S. H. Maeng, M. Essaid, and H. T. Ju, "Analysis of ethereum network properties and behavior of influential nodes," *2020 21st APNOMS IEEE*, pp. 203-20, 2020.

[12] S. H. Maeng, M. Essaid, C. Lee, S. Park, and H. Ju, "Visualization of Ethereum P2P network topology and peer properties," *Int. J. Netw. Manag.*, Jun. 2021.

[13] Y. Gao, J. Shi, X. Wang, Q. Tan, C. Zhao, and Z. Yin, "Topology measurement and analysis on ethereum P2P network," *2019 ISCC*, pp. 1-7, Jun. 2019.

[14] "*Ethereum Clinet*," https://ethereum.org/en/developers/docs/nodes-and-clients/, cited Feb. 2021.

[15] "*Ethereum Requirements*," https://ethereum.org/ko/developers/docs/nodes-and-clients/, cited Feb. 2021.

[16] "*Ethernodes*," https://www.ethernodes.org/, cited Feb. 2021.

[17] "*Etherscan Node Tracker*," https://etherscan.io/nodetracker, cited Feb. 2021.

[18] "*Ethereum devp2p*," https://github.com/ethereum/devp2p, cited Mar. 2021.

맹 수 훈 (Soohoon Maeng)

2019년 2월 : 계명대학교 컴퓨터공학과 졸업

2021년 8월 : 계명대학교 컴퓨터공학과 석사

2021년 9월~현재 : 대구경북첨단의료산업진흥재단

<관심분야> 네트워크, 블록체인, 의료 SW

[ORCID:0000-0002-0169-2817]

주 홍 택 (Hongtaek Ju)

1989년 2월 : KAIST 컴퓨터공학과 졸업

1991년 2월 : POSTECH 컴퓨터공학과 석사

2002년 2월 : POSTECH 컴퓨터공학과 박사

2002년 9월~현재 : 계명대학교 컴퓨터공학과 교수

<관심분야> 컴퓨터 공학, 네트워크, 블록체인

[ORCID:0000-0002-8434-485X]