(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국

(43) 국제공개일
2023 년 12 월 28 일 (28.12.2023)

WIPO | PCT

(10) 국제공개번호
WO 2023/249166 A1

(51) 국제특허분류:
H04L 67/1074 (2022.01) *H04L 67/1097* (2022.01)
H04L 43/0852 (2022.01) *H04L 12/18* (2006.01)
H04L 43/0876 (2022.01) *H04L 69/22* (2022.01)
H04L 67/1042 (2022.01)

(21) 국제출원번호: PCT/KR2022/015119

(22) 국제출원일: 2022 년 10 월 7 일 (07.10.2022)

(25) 출원언어: 한국어

(26) 공개언어: 한국어

(30) 우선권정보:
10-2022-0075510 2022 년 6 월 21 일 (21.06.2022) KR
10-2022-0085477 2022 년 7 월 12 일 (12.07.2022) KR

(71) 출원인: 포항공과대학교 산학협력단 (**POSTECH RESEARCH AND BUSINESS DEVELOPMENT FOUNDATION**) [KR/KR]; 37673 경상북도 포항시 남구 청암로 77, Gyeongsangbuk-do (KR).

(72) 발명자: **송황준 (SONG, Hwang Jun)**; 37673 경상북도 포항시 남구 청암로 77, Gyeongsangbuk-do (KR). **지승규 (JI, Seung Gyu)**; 37673 경상북도 포항시 남구 청암로 77, Gyeongsangbuk-do (KR). **노현민 (NOH, Hyun Min)**; 37673 경상북도 포항시 남구 청암로 77, Gyeongsangbuk-do (KR).

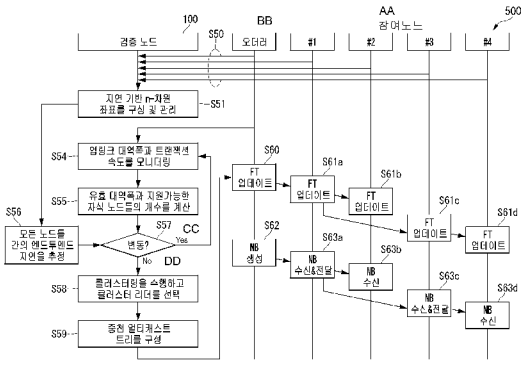
(74) 대리인: **특허법인이상 (E-SANG PATENT & TRADEMARK LAW FIRM)**; 06747 서울특별시 서초구 바우뫼로 188, 3층, Seoul (KR).

(81) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 국내 권리의 보호를 위하여): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 역내 권리의 보호를 위하여): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 유라시아 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 유럽 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME,

(54) **Title:** EFFECTIVE BANDWIDTH ESTIMATION METHOD FOR TRANSMITTING BLOCK DATA OF MULTI-CLOUD, CONTROL MESSAGE GENERATION METHOD, TREE INFORMATION TRANSMISSION METHOD, AND BLOCK TRANSMISSION SYSTEM FOR SAME

(54) **발명의 명칭:** 멀티 클라우드의 블록 데이터 전송을 위한 유효 대역폭 추정 방법과 제어 메시지 생성 방법 및 트리 정보 전달 방법과 이를 위한 블록 전송 시스템



100 ... Validator node
S60, S61a, S61b, S61c, S61d ... Update
S62 ... Generate
S63a, S63c ... Receive & transmit
S63b, S63d ... Receive
S51 ... Configure and manage delay-based n-dimensional coordinates
S54 ... Monitor uplink bandwidth and transaction speed
S55 ... Calculate effective bandwidth and number of supportable child nodes
S56 ... Estimate end-to-end delay among all nodes
S57 ... Is there change?
S58 ... Perform clustering and select cluster leader
S59 ... Configure nested multicast tree
AA ... Participant node
BB ... Orderer
CC ... Yes
DD ... No

(57) **Abstract:** A block transmission system for a multi-cloud, an effective bandwidth estimation method for transmitting block data, a control message generation method, and a tree information transmission method are disclosed. The effective bandwidth estimation method comprises the steps of: calculating, during a generation cycle of the block according to a Gaussian model block, transmission time for completing transmission of the previous block with a preset success probability before the next block is generated; and estimating an effective bandwidth on the basis of the number of transactions included in the block, the size of the transaction, the size of a block header, and the block transmission time.

(57) **요약서:** 멀티 클라우드의 블록 전송 시스템과 블록 데이터 전송을 위한 유효 대역폭 추정 방법과 제어 메시지 생성 방법 및 트리 정보 전달 방법이 개시된다. 유효 대역폭 추정 방법은, 가우시안 모델을 따르는 블록의 생성 주기에서, 다음 블록의 생성 전에 기설정된 성공 확률로 이전 블록의 전송을 완료하기 위한 블록 전송 시간을 계산하는 단계, 및 블록에 포함되는 트랜잭션 개수와 트랜잭션 크기와 블록 헤더 크기 및 블록 전송 시간에 기초하여 유효 대역폭을 추정하는 단계를 포함한다.

[다음 쪽 계속]

WO 2023/249166 A1

MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR),
OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM,
ML, MR, NE, SN, TD, TG).

공개:

— 국제조사보고서와 함께 (조약 제21조(3))

명세서

발명의 명칭: 멀티 클라우드의 블록 데이터 전송을 위한 유효 대역폭 추정 방법과 제어 메시지 생성 방법 및 트리 정보 전달 방법과 이를 위한 블록 전송 시스템

기술분야

[1] 본 발명은 멀티 클라우드의 확장가능한 하이퍼레저 패브릭 블록체인을 위한 블록 전송 시스템과 블록 전송 시스템에 적용할 수 있는 블록 데이터 전송을 위한 유효 대역폭 추정 방법, 제어 메시지 생성 방법 및 트리 정보 전달 방법에 관한 것이다.

배경기술

[2] 분산 원장인 블록체인 시스템은 저장된 데이터를 변경하고 속이는 것을 불가능하게 하여 학계와 산업계에서 상당한 주목을 받고 있다. 가장 인기 있는 블록체인 애플리케이션 중 하나는 암호화폐이다. 최근 온라인 시장에서 블록체인 기반의 대체 불가능 토큰(non-fungible token, NFT)이 급부상하면서 가치(value)의 새로운 패러다임을 만들고 있다. 다양한 서비스의 블록체인 애플리케이션은 높은 수준의 확장성 요구 사항을 제기하고 있다. 잘 알려진 블록체인 트릴레마(trilemma)는 본질적으로 분산 환경에서 안전하고 확장 가능한 블록체인을 설계하는 것이 불가능하다는 것이다. 따라서 블록체인 시스템은 통상 이러한 속성 중 하나를 희생하여 목적에 따라 설계된다.

[3] 일반적으로 블록체인 시스템은 탈중앙화를 희생하는 허가형 블록체인과 확장성을 희생하는 무허가형 블록체인의 두 가지 유형으로 나누어진다. 허가된 블록체인에서는 허가된 피어들만 합의에 참여할 수 있는 반면, 허가 없는 블록체인에서는 누구나 합의에 참여할 수 있습니다.

[4] 하이퍼레저 패브릭(hyperledger fabric)은 가장 널리 사용되는 허가된 블록체인 플랫폼 중 하나이다. 하이퍼레저 패브릭은 높은 확장성을 달성하기 위해 다양한 노드에서 수행되는 일련의 실행, 주문 및 유효성 검사를 통해 사용자 트랜잭션을 확인한다. 하이퍼레저 패브릭의 성능을 더욱 향상시키기 위해 실행과 합의 절차를 분리하기 위한 많은 연구 노력이 이루어지고 있다.

[5] 예를 들어, 하이퍼레저 패브릭의 높은 처리량(throughtput)을 위해 실행 및 주문의 순차적 워크플로가 병렬로 수행되는 병렬 워크플로가 개발되었다. 또한, 하이러레저 패브릭이 유효하지 않은 트랜잭션을 최소화하면서 동시 트랜잭션 실행을 사용하는 효과적인 하이브리드 트랜잭션 실행 접근 방식이 도입되었다.

[6] 그러나 하이퍼레저 패브릭의 높은 수준의 확장성에도 불구하고 분산 원장의 무결성, 간단한 합의 메커니즘 및 단일 데이터 센터나 클라우드의 시스템 배포와 같이 하이퍼레저 패브릭 블록체인의 취약한 보안 및 분산화에 대한 본질적인 우려가 있다.

[7] 지금까지, 허가된 블록체인에서도 보안 및 탈중앙화 문제를 다루기 위한 연구가 진행되어 왔다. 예를 들어, 중앙 집중식 서비스 공급업체에 의존하는 시스템의 보안 위협을 완화하기 위해 신뢰할 수 있는 BaaS(blockchain-as-a-service)를 위한 시행자 아키텍처가 제안되었다. 또한, 스마트 계약이 보안 스토리지 감사를 위해 모델링된 온체인 개인정보보호 문제(on-chain privacy concerns)를 해결하기 위해 블록체인 지원 감사 프레임워크(blockchain-enabled auditing framework)가 도입되었다.

[8] 또한, 하이퍼레저 패브릭 워킹 그룹은 서로 다른 클라우드에 호스팅된 서비스 공급업체가 인프라와 독립적으로 비즈니스 네트워크를 구축하고 가입할 수 있도록 하는 상호 운용성 워크플로(interoperability workflow)를 지정하고 표준화하려고 시도하고 있다. 그러나 다중 클라우드를 통한 패브릭 블록체인 아키텍처는 네트워크 지연으로 인해 단일 클라우드 환경에 비해 트랜잭션 처리량이 현저히 낮을 수 있고, 결국 확장성 문제가 발생할 수 있다.

[9] 게다가, 효율적인 P2P(Peer-to-Peer) 네트워크 프로토콜이 합의와 검증 프로세스 사이의 대기 시간을 줄임으로써 높은 트랜잭션 처리량을 달성할 수 있다는 사실이 이미 알려져 있다. 실제로 P2P 네트워크는 블록체인의 백본 네트워크에서 역할을 하기 때문에 P2P 네트워크의 설계는 블록체인의 확장성을 향상시키는 중요한 요소이다. 패브릭 및 이더리움(Ethereum) 블록체인은 통상 가십(Gossip) P2P 및 devP2P와 같은 P2P 네트워크 모듈을 구현한다.

[10] 전통적으로 P2P 네트워크 프로토콜은 트리 기반 기술과 메시 기반 기술의 두 그룹으로 분류할 수 있다. 트리 기반 기술에서 피어(Peer)에는 상위 피어와 여러 하위 피어가 있다. 피어는 상위 피어로부터 데이터를 수신하고 수신된 데이터를 하위 피어에게 중계한다. 예를 들어, 기존의 트리 기반 기술은 통상 소스에서 대상으로 데이터를 푸시하여 지연을 최소화할 수 있으므로 짧은 지연을 가진다. 그러나 피어 이탈 또는 패킷 손실이 발생하면 모든 하위 노드들이 서비스 품질(quality of service, QoS) 저하를 경험한다. 이러한 트리 기반 기술의 단점을 극복하기 위해 메쉬 기반 기술이 제안되었다.

[11] 메쉬 기반 기술에서 피어는 다른 피어에 대한 다중 링크를 가지며 상위-하위 관계는 없다. 피어들을 서로 데이터를 주고받고, 전달 메커니즘으로서 풀(pull) 전달을 채택한다. 기존 메시 기반 기술의 일례로는 쿨스트리밍(CoolStreaming), 불릿(Bullet), 체인소(Chainsaw), P2P 네트워크를 통한 IPTV, 메시 풀 기반 P2P 스트리밍 등이 있다.

[12] 이와 같이 패브릭 블록체인의 성능을 개선하기 위한 다양한 연구 노력에도 불구하고 아직까지 멀티 클라우드(multi-clouds) 상에서 패브릭 블록체인의 트랜잭션 처리량을 개선하기 위한 방안은 미흡한 실정이다.

발명의 상세한 설명

기술적 과제

[13] 본 발명은 전술한 종래 기술의 요구에 부응하기 위해 도출된 것으로, 본 발명의 목적은 멀티 클라우드에 무작위로 위치한 모든 참여 노드에 가능한 한 빨리 블록을 전달하여 멀티 클라우드에서 하이퍼레저 패브릭 블록체인의 확장성과 트랜잭션 처리량을 높일 수 있는 지능형 블록 전송 시스템을 제공하는데 있다.

[14] 본 발명의 다른 목적은 멀티 클라우드의 확장가능한 하이퍼레저 패브릭 블록체인을 위한 블록 전송 시스템에 적용할 수 있는 블록 데이터 전송을 위한 유효 대역폭 추정 방법 및 장치를 제공하는데 있다.

[15] 본 발명의 또 다른 목적은 멀티 클라우드에 무작위로 위치한 모든 참여 노드들에 가능한한 블록을 빠르게 전달하여 멀티 클라우드에서 하이퍼레저 패브릭 블록체인의 확장성과 트랜잭션 처리량을 높일 수 있는 블록 데이터 전송을 위한 유효 대역폭 추정 방법 및 장치를 제공하는데 있다.

[16] 본 발명의 또 다른 목적은 멀티 클라우드의 확장가능한 하이퍼레저 패브릭 블록체인을 위한 블록 전송 시스템의 제어 메시지 생성 방법 및 이를 위한 제어 메시지 포맷을 제공하는데 있다.

[17] 본 발명의 또 다른 목적은 멀티 클라우드에 무작위로 위치한 모든 참여 노드들에 가능한한 블록을 빠르게 전달하여 멀티 클라우드에서 하이퍼레저 패브릭 블록체인의 확장성과 트랜잭션 처리량을 높일 수 있는 지능형 블록 전송 시스템의 트리 정보 전달 방법을 제공하는데 있다.

[18] 본 발명의 또 다른 목적은 전술한 지능형 블록 전송 시스템에 적용할 수 있는 제어 메시지 포맷을 이용하는 P2P(peer to peer) 네트워크 프로토콜을 제공하는데 있다.

과제 해결 수단

[19] 상기 기술적 과제를 해결하기 위하여 본 발명에서는 단일 데이터 센터가 아닌 멀티 클라우드를 통해 패브릭 블록체인(fabric blockchain)의 트랜잭션 처리량을 개선하는 데 중점을 둔다. 즉, 본 발명의 블록 전송 시스템은 멀티 클라우드에서 패브릭 블록체인의 백본 네트워크 역할을 성공적으로 수행하기 위해 하이브리드 P2P(peer to peer) 멀티캐스트 트리 네트워크를 기반으로 P2P 시스템을 설계하고 패브릭의 P2P 네트워크 모듈과 통합하도록 구성된다.

[20] 한편, 클라우드 환경에서는 업로드 대역폭이 다운로드 대역폭보다 제한될 수 있다. 즉, 한 쌍의 노드는 P2P 네트워크를 통해 논리적으로 연결될 수 있지만 상대적으로 넓은 범위에서 지연이 변하고 네트워크 상황에 따라 연결이 깨지는 경우가 있다. 이는 효과적인 P2P 네트워크 구축에 큰 걸림돌이 된다. 이에 본 발명의 블록 전송 시스템의 블록 전송 프로세스에서는 주요 구성요소들로서 지연 추정 과정, 유효 대역폭 추정 과정, 노드 클러스터링 및 클러스터 리더 선정 과정, 하이브리드 P2P 멀티캐스트 트리 구축 과정을 포함한다.

[21] 특히, 본 발명은 멀티 클라우드의 확장 가능한 하이퍼레저 패브릭 블록체인을 위한 블록 전송 시스템에 적용할 수 있는 블록 데이터 전송을 위한 유효 대역폭

추정 방법 및 장치를 제공한다. 블록 데이터 전송을 위한 유효 대역폭(effective bandwidth) 추정은, 하이퍼레저 패브릭에서 효과적인 P2P 멀티캐스트 트리를 구성하기 위한 블록 전파에 필요한 유효 대역폭을 추정하는 것으로서, 트랜잭션이 기하급수적으로 분포하고 오더러에서 블록이 고정된 크기로 생성된다고 가정할 때, 블록이 생성되는 주기는 가우시안 모델을 따르고 다음 블록이 생성되기 전에 이전 블록의 전송을 일정 확률 이상에서 완료하기 위한 블록 전송 시간을 계산하고, 그 때의 유효 대역폭을 계산하도록 구성된다. 계산된 유효 대역폭은 블록에 포함된 트랜잭션 개수와 트랜잭션 발생 속도에 따라 증가할 수 있다.

[22] 상기 기술적 과제를 해결하기 위한 본 발명의 일 측면에 따른 멀티 클라우드에서의 블록 데이터 전송을 위한 유효 대역폭 추정 방법은, 멀티 클라우드에서의 블록 데이터 전송을 위한 유효 대역폭 추정 방법으로서, 가우시안 모델을 따르는 블록의 생성 주기에서, 다음 블록의 생성 전에 기설정된 성공 확률로 이전 블록의 전송을 완료하기 위한 블록 전송 시간을 계산하는 단계; 및 상기 블록에 포함되는 트랜잭션 개수와 트랜잭션 크기와 블록 헤더 크기 및 상기 블록 전송 시간에 기초하여 유효 대역폭을 추정하는 단계를 포함한다.

[23] 일실시예에서, 상기 계산하는 단계에서 상기 가우시안 모델을 따르는 블록의 생성 주기에서 상기 블록 데이터 전송을 위한 트랜잭션들은 기하급수적으로 분포하고, 상기 멀티 클라우드의 오더러(orderer)에서 블록이 고정된 크기로 생성되는 것으로 설정될 수 있다.

[24] 일실시예에서, 상기 방법은 미리 설정된 개수의 트랜잭션들이 오더러에 도착할 때 상기 블록을 생성하는 단계를 더 포함할 수 있다.

[25] 일실시예에서, 상기 계산하는 단계에서 상기 블록에 포함된 인접한 두 트랜잭션들 사이의 시간 간격은 기하급수적으로 분포하는 것으로 설정될 수 있다.

[26] 일실시예에서, 상기 방법은, 상기 블록에 포함된 트랜잭션 개수가 미리 설정된 기준값 이상으로 클 때, 상기 트랜잭션들이 도착할 때까지의 시간 간격을 상기 가우시안 모델로 정의되는 단계를 더 포함할 수 있다.

[27] 일실시예에서, 상기 가우시안 모델은 수학식 12와 같이 정의될 수 있다.

[28] 일실시예에서, 상기 블록 전송 시간은 수학식 13과 같이 정의될 수 있다.

[29] 일실시예에서, 상기 방법은 상기 블록 전송 시간이 더 작게 설정됨에 따라서 업로드 대역폭 사용을 희생하면서 더 작은 블록 버퍼링 시간을 확률적으로 보장하는 성공 확률을 설정하는 단계를 더 포함할 수 있다.

[30] 일실시예에서, 상기 유효 대역폭은 수학식 14와 같은 수식에 기초하여 추정될 수 있다.

[31] [수학식 14]

[32] $$BW_{eff} = \frac{N_{trx}^{blk} \cdot B_{TX} + B_{blk\_hd}}{\tilde{t}_{BLK}}$$

[33] 여기서, B_{TX}는 상기 트랜잭션 크기이고, B_{blk_hd}는 상기 블록 헤더 크기이며, $\tilde{t}_{BLK}$는 상기 트랜잭션들이 도착할 때까지의 시간 간격이다.

[34] 일실시예에서, 상기 트랜잭션 개수가 소정의 기준값보다 클 때, 상기 유효 대역폭은 상기 지수 분포 확률 모델의 트랜잭션 속도에 따라 선형적으로 증가한다.

[35] 일실시예에서, 상기 지수 분포 확률 모델의 트랜잭션 속도가 소정의 기준값보다 크면, 상기 유효 대역폭은 상기 트랜잭션 개수에 대해 기하급수적으로 감소한다.

[36] 일실시예에서, 상기 계산하는 단계에서 상기 성공 확률이 커질수록 상기 트랜잭션 개수의 기하급수적 감소율은 더 커지도록 설정될 수 있다.

[37] 일실시예에서, 상기 방법은, 상기 트랜잭션 개수가 미리 결정된 트랜잭션 개수보다 작더라도, 상기 트랜잭션들이 도착할 때까지의 시간 간격을 미리 설정된 크기보다 작게 유지하기 위하여 상기 트랜잭션들 중 가장 오래된 트랜잭션의 버퍼링 시간이 소정의 임계값보다 크면 그 즉시 블록을 생성하는 단계를 더 포함할 수 있다.

[38] 상기 기술적 과제를 해결하기 위한 본 발명의 일 측면에 따른 멀티 클라우드에서의 블록 데이터 전송을 위한 유효 대역폭 추정 장치는, 멀티 클라우드에서의 블록 데이터 전송을 위한 유효 대역폭 추정 장치로서, 프로세서; 및 상기 프로세서에 연결되고 상기 프로세서에 의해 실행되는 적어도 하나의 명령을 저장하는 메모리를 포함하고, 상기 적어도 하나의 명령은, 상기 프로세서에 의해, 가우시안 모델을 따르는 블록의 생성 주기에서, 다음 블록의 생성 전에 기설정된 성공 확률로 이전 블록의 전송을 완료하기 위한 블록 전송 시간을 계산하는 단계, 및 상기 블록에 포함되는 트랜잭션 개수와 트랜잭션 크기와 블록 헤더 크기 및 상기 블록 전송 시간에 기초하여 유효 대역폭을 추정하는 단계를 수행하도록 구성될 수 있다.

[39] 일실시예에서, 상기 적어도 하나의 명령은, 상기 프로세서에 의해, 상기 계산하는 단계에서 상기 가우시안 모델을 따르는 블록의 생성 주기에서 상기 블록 데이터 전송을 위한 트랜잭션들은 기하급수적으로 분포하고 상기 멀티 클라우드의 오더러(orderer)에서 블록을 고정된 크기로 생성하도록 처리할 수 있다.

[40] 일실시예에서, 상기 적어도 하나의 명령은, 상기 프로세서에 의해, 미리 설정된 개수의 트랜잭션들이 오더러에 도착할 때 상기 블록을 생성하는 단계를 더 수행하도록 할 수 있다.

[41] 일실시예에서, 상기 적어도 하나의 명령은, 상기 프로세서에 의해, 상기 블록에 포함된 인접한 두 트랜잭션들 사이의 시간 간격을 기하급수적으로 분포하는

것으로 처리할 수 있다.

[42] 일실시예에서, 상기 적어도 하나의 명령은, 상기 프로세서에 의해, 상기 블록에 포함된 트랜잭션 개수가 미리 설정된 기준값 이상으로 클 때, 트랜잭션들이 도착할 때까지의 시간 간격을 상기 가우시안 모델로 정의할 수 있다.

[43] 일실시예에서, 상기 적어도 하나의 명령은, 상기 프로세서에 의해, 상기 블록 전송 시간이 더 작게 설정됨에 따라서 업로드 대역폭 사용을 희생하면서 더 작은 블록 버퍼링 시간을 확률적으로 보장하도록 상기 성공 확률을 설정하는 단계를 더 수행하도록 구성될 수 있다.

[44] 일실시예에서, 상기 적어도 하나의 명령은, 상기 프로세서에 의해, 상기 블록에 포함된 트랜잭션들의 개수가 미리 결정된 트랜잭션 개수보다 작더라도, 상기 트랜잭션들이 도착할 때까지의 시간 간격을 미리 설정된 크기보다 작게 유지하기 위하여, 상기 트랜잭션들 중 가장 오래된 트랜잭션의 버퍼링 시간이 소정의 임계값보다 크면 그 즉시 블록을 생성하는 단계를 더 수행하도록 구성될 수 있다.

[45] 또한, 본 발명은 멀티 클라우드의 확장 가능한 하이퍼레저 패브릭 블록체인을 위한 블록 전송 시스템에 적용할 수 있는 제어 메시지 포맷과 이러한 제어 메시지 포맷을 이용하는 트리 정보 전달 프로세스를 제공한다. 본 발명의 제어 메시지 포맷은 제어 메시지 포맷을 생성하거나, 저장하거나, 전송하는 방법이나 장치에 대한 것으로, 이러한 방법 및 장치를 포함할 수 있다.

[46] 상기 기술적 과제를 해결하기 위한 본 발명의 또 다른 측면에 따른 제어 메시지 생성 방법은, 멀티 클라우드의 확장 가능한 하이퍼레저 패브릭 블록체인을 위한 블록 전송 시스템의 제어 메시지 생성 방법으로서, 복수의 참여 노드들을 복수의 클러스터들에 속한 노드들로 구분하는 단계; 상기 복수의 참여 노드들 중에서 상기 복수의 클러스터들 각각의 클러스터 리더를 선택하는 단계; 및 상기 복수의 클러스터들과 상기 클러스터 리더에 대한 정보에 기초하여 하이브리드 P2P(peer to peer) 멀티캐스트 트리(multicast tree)를 구성하는 단계를 포함한다.

[47] 일실시예에서, 상기 구성하는 단계는, 상기 하이브리드 P2P 멀티캐스트 트리에 포함되는 각 참여 노드의 트리 정보를 노드 IP(internet protocol) 필드, 자식 노드 수(the number of child nodes) 필드 및 자식 노드 목록의 오프셋(offset of child nodes lists) 필드의 조합으로 형성하는 단계를 포함할 수 있다.

[48] 일실시예에서, 상기 노드 IP 필드는 현재 노드의 IP 주소를 나타내고, 상기 자식 노드 수 필드는 상기 하이브리드 P2P 멀티캐스트 트리를 통해 상기 현재 노드에 연결된 자식 노드의 수를 나타내고, 상기 자식 노드 목록의 오프셋 필드는 제어 메시지 자체 내에서 첫 번째 자식 노드의 위치를 나타낼 수 있다.

[49] 일실시예에서, 상기 구성하는 단계는, 상기 각 참여 노드의 트리 정보를 미리 설정된 순서대로 일렬로 배열한 페이로드를 형성하는 단계를 더 포함할 수 있다.

[50] 일실시예에서, 상기 제어 메시지 생성 방법은, 상기 복수의 참여 노드들 중 상기 페이로드를 포함한 제어 메시지를 받은 특정 참여 노드가 하위 트리에 대한

정보를 제거하여 상기 제어 메시지를 재구성하는 단계를 더 포함할 수 있다.

[51] 일실시예에서, 상기 제어 메시지 생성 방법은, 상기 복수의 참여 노드들의 각 노드와 상기 멀티 클라우드의 오더러(orderer) 간에 추정된 엔드투엔드(end-to-end) 지연(delay)에 기초하여 획득되는 상기 복수의 참여 노드들의 유효 대역폭과 지원가능한 자식 노드들의 개수 중 적어도 어느 하나가 미리 설정된 기준 범위를 벗어난 변동이 있는지를 판단하는 단계를 더 포함할 수 있다. 여기서, 상기 구성하는 단계는 상기 판단하는 단계의 판단 결과가 변동 있음인 경우에 수행될 수 있다.

[52] 일실시예에서, 상기 제어 메시지 생성 방법은, 상기 구분하는 단계, 상기 선택하는 단계 및 상기 구성하는 단계를 반복적으로 수행하여 클러스터 개수와 클러스터 리더를 최종 결정하는 단계를 더 포함할 수 있다.

[53] 상기 기술적 과제를 해결하기 위한 본 발명의 다른 측면에 따른 제어 메시지 포맷은, 전술한 제어 메시지 생성 방법의 실시예들 중 어느 하나의 제어 메시지 생성 방법에 이용되는 제어 메시지 포맷으로서, 헤더 및 페이로드를 포함하고, 상기 헤더는 버전(version), 메시지 유형(message type) 및 메시지 길이(message length)에 대한 정보를 담고, 상기 페이로드는 소정의 순번들에 따라 복수의 노드들의 트리 정보를 각각 순차적으로 나열한 형태를 구비한다.

[54] 일실시예에서, 상기 복수의 노드들 각각의 트리 정보는 노드 IP(internet protocol), 자식 노드 수(the number of child nodes) 및 자식 노드 목록의 오프셋(offset of child nodes lists)에 대한 필드들을 구비할 수 있다.

[55] 상기 기술적 과제를 해결하기 위한 본 발명의 또 다른 측면에 따른 트리 정보 전달 방법은, 멀티 클라우드의 확장가능한 하이퍼레저 패브릭 블록체인을 위한 블록 전송 시스템의 트리 정보 전달 방법으로서, 멀티 클라우드에 연결된 검증 노드에서 모든 참여 노드들로부터 좌표 값들을 수신하는 단계; 오더러(orderer)로부터 트랜잭션 속도(transaction rate)를 수신하는 단계; 상기 모든 참여 노드들 중 상기 오더러에 연결되는 앵커 노드들의 업링크 대역폭을 추정하는 단계; 상기 모든 참여 노드들의 유효 대역폭과 지원가능한 자식 노드들의 개수를 계산하는 단계; 상기 모든 참여 노드들의 각 참여 노드와 상기 오더러 간의 엔드투엔드 지연을 추정하는 단계; 상기 엔드투엔드 지연에 기초하여 상기 유효 대역폭과 상기 지원가능한 자식 노드들의 개수가 미리 설정된 기준 범위를 벗어하는 변동이 있는지를 판단하는 단계; 상기 변동이 있는 경우, 상기 좌표 값들과 상기 트랜잭션 속도에 대한 정보에 기초하여 상기 모든 참여 노드들을 복수의 클러스터들로 클러스터링하는 단계; 상기 복수의 클러스터들 각각의 클러스터 리더를 선택하는 단계; 상기 복수의 클러스터와 상기 클러스터 리더를 토대로 중첩 멀티캐스트 트리를 구성하는 단계; 및 상기 중첩 멀티캐스트 트리를 제어 메시지에 담아 상기 오더러로 전송하는 단계를 포함한다.

[56] 일실시예에서, 상기 제어 메시지는 상기 모든 참여 노드들에 전달되어 각 참여

노드에서 메시지 전달 테이블의 업데이트에 이용될 수 있다.

[57] 일실시예에서, 상기 트리 정보 전달 방법은, 상기 클러스터링하는 단계, 상기 클러스터 리더를 선택하는 단계 및 상기 중첩 멀티캐스트 트리를 구성하는 단계를 반복적으로 수행하여 클러스터 개수와 클러스터 리더를 최종적으로 결정하는 단계를 더 포함할 수 있다.

[58] 일실시예에서, 상기 트리 정보 전달 방법은, 상기 복수의 참여 노드들 중 상기 페이로드를 포함한 제어 메시지를 받은 특정 참여 노드가 일부 하위 트리에 대한 정보를 제거하여 상기 제어 메시지를 재구성하는 단계를 더 포함할 수 있다.

[59] 일실시예에서, 상기 제어 메시지의 크기는 상기 중첩 멀티캐스트 트리를 통해 상기 오더러로부터 리프 노드들까지 전달되는 동안 각 부모 노드에서 단계적으로 감소될 수 있다.

발명의 효과

[60] 전술한 멀티 클라우드 기반 확장 가능한 하이퍼레저 패브릭 블록체인을 위한 지능형 블록 전송 시스템을 사용하면, 노드들 간 구성된 하이브리드 P2P(peer to peer) 멀티캐스트 트리를 통해 트리 정보가 담긴 제어 메시지를 전파함으로써 각 클라우드의 노드들의 트리 정보를 적은 오버헤드로 전파할 수 있고, 그에 의해 하이퍼레저 패브릭 블록체인의 확장성과 트랜잭션 처리량을 향상시킬 수 있다.

[61] 또한, 본 발명에 의하면, 블록 전송 시스템에서 모든 참여 노드에 하이브리드 P2P 멀티캐스트 트리 정보를 효율적으로 제공하기 위해 블록 데이터 전송을 위한 유효 대역폭 추정 방안이나 이를 위한 모델링 방법을 제공할 수 있다.

[62] 또한, 본 발명에 의하면, 블록 전송 시스템에서 모든 참여 노드에 하이브리드 P2P 멀티캐스트 트리 정보를 효율적으로 제공하기 위한 제어 메시지 형식과 전달 프로세스를 제공할 수 있다.

도면의 간단한 설명

[63] 도 1은 본 발명의 일실시예에 따른 블록 전송 시스템을 적용할 수 있는 멀티 클라우드 환경에 대한 예시도이다.

[64] 도 2는 본 발명의 일실시예에 따른 블록 전송 시스템의 주요 구성요소들을 설명하기 위한 도면이다.

[65] 도 3은 도 2의 블록 전송 시스템에 채용할 수 있는 네트워크 모니터링 관리 모듈(network monitoring & management module)의 상세 구성에 대한 개략적인 블록도이다.

[66] 도 4는 도 2의 블록 전송 시스템에 채용할 수 있는 블록 전송 모듈(block transmission module)의 상세 구성에 대한 개략적인 블록도이다.

[67] 도 5는 도 2의 블록 전송 시스템의 전체적인 작동 절차에 대한 흐름도이다.

[68] 도 6 내지 도 11은 도 2의 블록 전송 시스템에 채용할 수 있는 0(n) 제어 메시지를 이용한 지연 추정 프로세스를 설명하기 위한 도면들이다.

[69] 도 12 내지 도 15는 도 2의 블록 전송 시스템에 채용할 수 있는 블록 전송을 위한

유효 대역폭 추정 프로세스를 설명하기 위한 도면들이다.

[70] 도 16은 도 2의 블록 전송 시스템에 채용할 수 있는 트리 정보 전달을 위한 제어 메시지 포맷에 대한 예시도이다.

[71] 도 17 및 도 18은 도 16의 제어 메시지를 이용한 트리 구조 식별 과정을 설명하기 위한 예시도들이다.

[72] 도 19는 도 1의 블록 전송 시스템이 적용될 수 있는 멀티 클라우드의 네트워크 토폴로지에 대한 예시도이다.

[73] 도 20은 도 19의 멀티 클라우드 네트워크에 채용할 수 있는 멀티 클라우드 테스트베드의 시스템 아키텍처에 대한 예시도이다.

[74] 도 21은 도 2의 블록 전송 시스템에서의 지연 추정 프로세스에 대한 성능 검증 결과를 나타낸 그래프이다.

[75] 도 22a 내지 도 22c는 도 21의 지연 추정 프로세스와 관련하여 노드#0에서 모든 노드들까지 측정된 지연과 추정된 지연을 비교하여 나타낸 그래프들이다.

[76] 도 23 및 도 24는 도 2의 블록 전송 시스템에서의 클러스터링 및 하이브리드 P2P 멀티캐스트 트리 컨스트럭팅 프로세스(clustering & hybrid P2P multicast tree constructing processes, CCP)에 대한 성능 검증 결과를 나타낸 그래프들이다.

[77] 도 25 및 도 26은 도 23 및 도 24의 CCP와 관련하여 업링크 대역폭이 200Mbps일 때의 평균 블록 도착 시간과 최대 블록 도착 시간의 지연 비교를 위한 그래프들이다.

[78] 도 27은 본 발명의 다른 실시예에 따른 블록 전송 시스템의 구성에 대한 개략적인 블록도이다.

발명의 실시를 위한 형태

[79] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[80] 제1, 제2 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.

[81] 본 출원의 실시예들에서, 'A 및 B 중에서 적어도 하나'는 'A 또는 B 중에서 적어도 하나' 또는 'A 및 B 중 하나 이상의 조합들 중에서 적어도 하나'를 의미할 수 있다. 또한, 본 출원의 실시예들에서, 'A 및 B 중에서 하나 이상'은 'A 또는 B

중에서 하나 이상' 또는 'A 및 B 중 하나 이상의 조합들 중에서 하나 이상'을 의미할 수 있다.

[82] 어떤 구성요소가 다른 구성요소에 '연결되어' 있다거나 '접속되어' 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 '직접 연결되어' 있다거나 '직접 접속되어'있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다.

[83] 본 출원에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, '포함한다' 또는 '가진다' 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[84] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가지고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥 상 가지는 의미와 일치하는 의미를 가진 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[85] 이하, 첨부한 도면들을 참조하여, 본 발명의 바람직한 실시예를 보다 상세하게 설명하고자 한다. 본 발명을 설명함에 있어 전체적인 이해를 용이하게 하기 위하여 도면상의 동일한 구성요소에 대해서는 동일한 참조부호를 사용하고 동일한 구성요소에 대해서 중복된 설명은 생략한다.

[86] 도 1은 본 발명의 일실시예에 따른 블록 전송 시스템을 적용할 수 있는 멀티 클라우드(multi-clouds) 환경에 대한 예시도이다.

[87] 도 1을 참조하면, 멀티 클라우드 환경은 제1 클라우드(Cloud A), 제2 클라우드(Cloub B), 제3 클라우드(Cloud C) 및 제4 클라우드(Cloud D)가 인터넷(internet)에 연결된 게이트웨이들(700)에 의해 서로 연결된 구조를 구비할 수 있다. 각 클라우드는 적어도 하나 이상의 참여 노드(participating node, 500)을 포함하고, 이 클라우드들 중 적어도 어느 하나의 클라우드 예컨대 제1 클라우드(Cloud A)는 검증 노드(validation node, 100)와 블록 생성 노드(block generating node)를 포함할 수 있다. 블록 생성 노드는 오더러(orderer, 300)로 지칭될 수 있고, 하이퍼레저 패브릭 블록체인 플랫폼으로도 지칭될 수 있다. 멀티 클라우드에 참여하는 참여 노드들(500)은 랜덤한 패턴으로 각 클라우드에 배치될 수 있다.

[88] 본 실시예에서 멀티 클라우드 환경에 채용할 수 있는 블록 전송 시스템은

블록체인 서비스에 사용되는 블록 데이터를 빠르게 모든 참여 노드들에 전달하여 멀티 클라우드에서 초당 트랜잭션 수(transactions per second, TPS)를 높일 수 있도록 구성된다.

[89] 도 2는 본 발명의 일실시예에 따른 블록 전송 시스템의 주요 구성요소들을 설명하기 위한 도면이다.

[90] 도 2를 참조하면, 블록 전송 시스템은 네트워크 모니터링 관리 모듈(network monitoring & management module, 110), 블록 생성 모듈(block generation module, 310), 및 블록 전송 모듈(block transmission module, 510)을 구비할 수 있다.

[91] 네트워크 모니터링 관리 모듈(110)은 검증 노드(100)에 탑재될 수 있고, 블록 생성 모듈(310)은 블록 생성 노드(300)에 탑재될 수 있고, 블록 전송 모듈(510)은 참여 노드(500)에 탑재될 수 있다. 또한, 검증 노드(100)나 블록 생성 노드(300)에는 블록 전송 모듈(510)과 동등 혹은 유사한 모듈이 탑재될 수 있다.

[92] 네트워크 모니터링 관리 모듈(110)과 블록 생성 모듈(310)과 블록 전송 모듈(510)은 이들 각각에 결합된 응용 프로그래밍 인터페이스(application programming interface, API)를 통해 서로 통신할 수 있다.

[93] 네트워크 모니터링 관리 모듈(110)과 블록 전송 모듈(510)은 제1 제어 메시지(control message #1)을 서로 주고 받을 수 있다. 제1 제어 메시지는 트리 구성을 위한 제어 메시지일 수 있다. 네트워크 모니터링 관리 모듈(110)을 좀더 상세히 설명하면 다음과 같다.

[94] 도 3은 도 2의 블록 전송 시스템에 채용할 수 있는 네트워크 모니터링 관리 모듈(network monitoring & management module)의 상세 구성에 대한 개략적인 블록도이다.

[95] 도 3을 참조하면, 네트워크 모니터링 관리 모듈(110)은 블록체인 인터페이스(120), 네트워크 모니터링 서브모듈(130), 네트워크 토폴로지 구성 서브모듈(140)을 구비할 수 있다. 네트워크 모니터링 관리 모듈(110)은 API(150)를 통해 자신이 탑재된 노드의 하드웨어와 연결되거나 외부의 컴퓨터나 컴퓨터 프로그램과 연결될 수 있다.

[96] 블록체인 인터페이스(120)는 블록체인 네트워크에 참여하는 노드들과 데이터를 주고 받기 위한 수단이나 이러한 수단에 상응하는 기능을 수행하는 구성부를 포함할 수 있다.

[97] 네크워크 모니터링 서브모듈(130)은 O(n) 오버헤드로 네트워크를 모니터링한다. 여기서 O(n)은 빅오 표기법(big-O notation)의 하나로서 알고리즘의 시간 복잡도를 나타낸다. 즉, 알고리즘의 복잡도를 판단하는 척도로는 시간 복잡도와 공간 복잡도가 있는데, O(n)은 알고리즘 내 연산의 횟수나 초당 트랜잭션 수와 관계가 있다.

[98] 네트워크 토폴로지 구성 서브모듈(140)은 하이브리드 P2P 멀티캐스트 트리(hybrid P2P multicast tree) 구조의 네트워크 토폴로지 컨스트럭팅(network topology constructing)을 수행한다.

[99] 도 4는 도 2의 블록 전송 시스템에 채용할 수 있는 블록 전송 모듈(block transmission module)의 상세 구성에 대한 개략적인 블록도이다.

[100] 도 4를 참조하면, 블록 전송 모듈(510)은 블록 데이터 송신기(520), 블록 데이터 수신기(530) 및 메시지 전달 테이블(message forwarding table, 550)을 구비할 수 있다. 블록 데이터 송신기(520) 및 블록 데이터 수신기(530)은 블록 데이터 송수신기(540)로 형성될 수 있다. 블록 전송 모듈(510)은 API(560)를 통해 자신이 탑재된 노드의 하드웨어와 연결되거나 외부의 컴퓨터나 컴퓨터 프로그램과 연결될 수 있다.

[101] 블록 데이터 송수신기(540)는 후술하는 제어 메시지 포맷으로 구성된 메시지를 포함한 블록 데이터를 블록체인 네트워크 또는 멀티 네트워크 상의 참여 노드들과 주고 받도록 구성된다.

[102] 메시지 전달 테이블(550)은 본 실시예에 따른 제어 메시지 포맷으로 구성된 제어 메시지 또는 이러한 제어 메시지를 포함한 메시지를 후술하는 멀티캐스트 크리 구조에 기반하여 전달하기 위해 설정되거나 업데이트될 수 있다.

[103] 도 5는 도 2의 블록 전송 시스템의 전체적인 작동 절차에 대한 흐름도이다.

[104] 도 5를 참조하면, 블록 전송 시스템에서 검증 노드(100)는 모든 참여 노드들(paticipating nodes #1, #2, #3, #4)로부터 좌표 값들을 수신하고 오더러(orderer)로부터 트랜잭션 속도(transaction rate)를 수신한다(S50). 이러한 정보에 기초하여 검증 노드(100)는 클러스터링 프로세스(clustering process)와 하이브리드 P2P 멀티캐스트 트리 컨스트럭팅 프로세스(hybrid P2P multicast tree constructing process)를 실행한다(S51 내지 S59). 그리고, 검증 노드(100)는 오더러와 모든 참여 노드들에 제어 메시지(control message)를 통해 트리 정보(tree information)를 전달한다.

[105] 그런 다음, 오더러는 트리 정보를 토대로 메시지 전달 테이블을 업데이트하고 업데이트한 메시지 전달 테이블을 적어도 하나의 특정 참여 노드로 전달한다(S60). 메시지 전달 테이블(message forwarding table)은 간략히 전달 테이블(forwarding table, FT)로 지칭될 수 있다. 또한, 오더러는 트리 정보를 토대로 새로운 블록(new block, NB)을 생성하고 생성한 블록들(blocks)을 트리 정보에 기초하여 적어도 하나의 다른 참여 노드로 전송할 수 있다(S62).

[106] 참여 노드들(paticipating nodes #1, #2, #3, #4)은 부모 노드와 자식 노드의 관계를 가질 수 있고, 부모 노드는 자신의 위치에서 적어도 하나의 자식 노드를 포함하는 트리 구조로 메시지 전달 테이블을 업데이트하고, 업데이트된 메시지 전달 테이블을 자식 노드로 전달할 수 있다(S61a 내지 S61d).

[107] 또한, 적어도 하나의 부모 노드는 오더러로부터 새로운 블록(NB)을 받고, 그 자식 노드로 새로운 블록을 전달할 수 있다(S63a 내지 S63d). 즉, 적어도 하나의 참여 노드는 중계 노드 형태로서 부모 노드와 자식 노드 모두로 기능할 수 있고, 적어도 하나의 다른 참여 노드는 자식 노드로서만 기능할 수 있다.

[108] 전술한 클러스터링 및 하이브리드 P2P 멀티캐스트 트리 컨스트럭팅

프로세스(hybrid P2P multicast tree constructing processes, CCP)를 좀더 상세히 설명하면 다음과 같다.

[109] 먼저 검증 노드(100)는 오더러와 모든 참여노드들로부터 수신한 정보를 토대로 지연 기반 n-차원 좌표를 구성하고 관리한다(S51).

[110] 다음, 검증 노드(100)는 오더러로부터 받은 트랜잭션 속도에 대한 제1 정보와 모든 참여 노드들로부터 받은 좌표 값들에 대한 제2 정보에 기초하여 업링크 대역폭과 트랜잭션 속도를 모니터링한다(S54).

[111] 다음, 검증 노드(100)는 유효 대역폭과 지원가능한 자식 노드들의 개수를 계산한다(S55).

[112] 다음, 검증 노드(100)는 지연 기반 n-차원 좌표 값들의 관리를 통해 모든 참여 노드들 사이의 엔드 투 엔드(end-to-end) 지연을 추정한다(S56).

[113] 다음, 추정된 엔드 투 엔드(end-to-end) 지연에 기초하여, 유효 대역폭과 지원가능한 자식 노드들의 개수 중 적어도 하나 이상이 미리 설정된 기준 범위를 벗어나거나 미리 설정된 오차 범위를 벗어나는 변동 이벤트가 발생하였는지를 판단한다(S57). 변동 이벤트가 발생하지 않은 경우, 검증 노드(100)는 현재 프로세스의 판단 단계에서 업링크 대역폭과 트랜잭션 속도를 모니터링하는 단계로 되돌아갈 수 있다.

[114] 다음, 변동 이벤트가 발생한 경우, 검증 노드(100)는 모든 참여 노드들에 대해 복수의 클러스터들로 클러스터링을 수행하고 각 클러스터의 클러스터 리더를 선택한다(S58).

[115] 다음, 검증 노드(100)는 복수의 클러스터들을 토대로 중첩 멀티캐스트 트리를 구성한다(S59). 그리고 검증 노드(100)는 오더러에 트리 정보(tree information)를 담은 제어 메시지(control message)를 전달한다. 이러한 제어 메시지는 모든 참여 노드들에 전달되어 각 참여 노드에서 메시지 전달 테이블의 업데이트에 이용될 수 있다.

[116] 도 6 내지 도 11은 도 2의 블록 전송 시스템에 채용할 수 있는 0(n) 제어 메시지를 이용한 지연 추정 프로세스를 설명하기 위한 도면들이다.

[117] 본 실시예에서 설명되는 두 노드들은 P2P 네트워크에서 논리적으로 서로 연결될수 있다. 따라서 모든 참여 노드들 사이의 종단(end-to-end) 간 지연 값을 추정하기 위한 모든 참여 노드들의 개수가 소정 개수(n_{nd})일 때 이론적으로 $O(n^2_{nd})$ 제어 메시지 오버헤드가 발생할 수 있고, 그에 의해 모든 참여 노드들의 개수만큼 네트워크 트랜잭션의 무게 혹은 개수가 증가하게 된다. 이에 본 실시예에서는 $O(n^2_{nd})$ 메시지 오버헤드에서 효과적인 지연 추정 프로세스를 제공한다.

[118] 이를 위해, 본 실시예에서는 각 노드가 지연 값을 기준으로 3차원 좌표에 배치되고, 3차원 벡터 공간에서 두 노드들 사이의 유클리드 거리가 지연 값을 나타내도록 구성된다. 이를 구체적으로 예시하면 다음과 같다. 이하의 예시의 프로세스에서 각 단계는 오더러와 연동하는 검증 노드에 의해 수행되거나 또는

검증 노드에 결합하는 지능형 블록 전송 시스템에 의해 수행될 수 있다. 지능형 블록 전송 시스템은 검증 노드를 포함할 수 있다.

[119] 먼저, 검증 노드는 하이퍼레저 패브릭(hyperledger fabric)의 오더러(orderer) 및 무작위로 선택된 앵커 노드들(anchor nodes)과 제어 메시지들을 교환함으로써 오더러와 앵커 노드를 사이에 3차원 좌표들이 구성된다. 3차원 좌표들의 구성은 아래의 제1 단계 내지 제5 단계를 참조할 수 있다. 앵커 노드는 참여 노드들에서 오더러와 연결되는 노드를 지칭할 수 있다.

[120] 그런 다음, 새로운 참여 노드의 좌표 값을 찾을 수 있다(제6 단계). 즉, 새로운 참여 노드는 오더러 및 앵커 노드들로 요청 메시지들을 보내고 이들의 좌표 값을 포함한 대응 응답 메시지들(corresponding reply messages)를 받고 이와 동시에 지연 값을 측정함으로써 자신의 좌표 값을 찾을 수 있습니다.

[121] 마지막으로, 참여 노드들의 좌표들의 정확도를 유지하기 위해 자체 교차 검사 알고리즘(self-cross check algorithm)을 수행할 수 있다(제7 단계).

[122] 전술한 제어 메시지 교환에 따라 3차원 좌표들을 구성하는 과정을 나타내면 다음의 제1 단계 내지 제5 단계와 같다. 제1 단계 내지 제5 단계에서, $d_{i,j}$는 노드 #i와 노드 #j 사이에 측정된 왕복 시간(round-trip-time, RTT)을 나타내고, 노드 #0(Node #0)은 하이퍼레저 패브릭(hyperledger fabric)의 오더러(orderer)를 나타낸다. 사실, 시변 지연은 좌표의 불확실성을 유발할 수 있다. 이 효과를 줄이기 위해서 오더러와 다른 앵커 노드들과 최대한 멀리 떨어진 참여 노드들 중에서 앵커 노드를 선택할 수 있다.

[123] 제1 단계에서는 오더러의 좌표를 (0,0,0)으로 고정한다. 이하 오더러는 노드#0(Node #0) 외에 원점 노드 또는 기준 노드로 지칭될 수 있다.

[124] 제2 단계에서는 오더러와 제1 앵커 노드를 연결하는 라인을 x-축으로서 정의한다. 즉, 제1 앵커 노드의 좌표는 도 6에 도시한 바와 같이 $(x_1,0,0)$으로 설정될 수 있다. 여기서, x_1은 원점(0,0.0)과의 간격에 대응하는 제1 거리($d_{0,1}$)의 값을 가질 수 있다. 제1 앵커 노드는 노드#1(Node #1)뿐 아니라 제1 노드로도 지칭될 수 있다.

[125] 제3 단계에서는 도 7에 도시한 바와 같이 오더러와 제1 앵커 노드를 연결하는 라인 상에 위치하지 않는 참여 노드들 중에서 제2 앵커 노드를 선택한다. 이 경우, 오더러의 좌표와 제1 앵커 노드의 좌표와 제2 앵커 노드의 좌표를 포함하는 평면을 x-y 평면으로 구성할 수 있다. 제2 앵커 노드는 노드#2(Node #2) 또는 제2 노드로 지칭될 수 있고, 그 좌표는 $(x_2,y_2,0)$로 표현될 수 있다. 여기서, 노드#2와 노드#0과의 제2 거리를 $d_{0,2}$로 정의하고, 노드#2와 노드#1과의 제3 거리를 $d_{1,2}$로 정의할 수 있다.

[126] 검증 노드는 해당 노드에서 측정된 RTT 값과 오더러 및 제1 앵커 노드로부터 받은 좌표 값들을 기반으로 다음의 수학식 1 및 수학식 2를 얻을 수 있다.

[127] [수식1]

$$\sqrt{x_2^2 + y_2^2} = d_{0,2}$$

[128] [수식2]

$$\sqrt{\left(x_2 - d_{0,1}\right)^2 + y_2^2} = d_{1,2}$$

[129] 수학식 1 및 수학식 2에서 (x_2,y_2)는 제2 앵커 노드의 좌표이다. 이에 의하면, 제2 앵커 노드의 좌표 값은 수학식 3 및 수학식 4에 의해 결정될 수 있다.

[130] [수식3]

$$x_2 = \frac{d_{0,2}^2 + d_{0,1}^2 - d_{1,2}^2}{2d_{0,1}}$$

[131] [수식4]

$$y_2 = \pm\sqrt{d_{0,2}^2 - \left(\frac{d_{0,2}^2 + d_{0,1}^2 - d_{1,2}^2}{2d_{0,1}}\right)^2}$$

[132] 수학식 3과 수학식 4에서 알 수 있듯이, 실제로 두 개의 좌표 값들이 수학식 1과 수학식 2를 만족한다. 본 실시예에서는 하기의 제5 단계에서 두 개의 좌표 값들 중 하나를 선택하도록 구성된다.

[133] 제4 단계에서는 도 8에 도시한 바와 같이 x-y 평면에 위치하지 않는 참여 노드들 중에서 제3 앵커 노드(the third anchor node)를 선택한다. 따라서, 소정의 x-y-z 공간이 오더러, 제1 앵커 노드, 제2 앵커 노드 및 제3 앵커 노드로 구성된다. 제3 앵커 노드는 노드#3(Node #3) 또는 제3 노드로 지칭될 수 있고, 그 좌표는 (x_3,y_3,z_3)로 표현될 수 있다. 여기서, 노드#3과 기준 노드와의 제4 거리를 $d_{0,3}$으로 정의하고, 노드#3과 노드#1과의 제5 거리를 $d_{1,3}$으로 정의하고, 노드#3과 노드#2와의 제6 거리를 $d_{2,3}$으로 정의할 수 있다.

[134] 또한, 검증 노드는 도 8에서와 같이 오더러, 제1 앵커 노드, 제2 앵커 노드 및 제3 앵커 노드에 대해 RTT 값들을 측정하거나 측정된 값을 수신한 후, 제3 앵커 노드의 후보 좌표 값들을 수학식 5 내지 수학식 7을 이용하여 계산할 수 있다.

[135] [수식5]

$$x_3 = \frac{d_{0,1}^2 + d_{0,3}^2 - d_{1,3}^2}{2d_{0,1}}$$

[136] [수식6]

$$y_3 = \frac{d_{0,3}^2 - 2x_2x_3 + y_2^2 - d_{2,3}^2 + x_2^2}{2y_2}$$

[137] [수식7]

$$z_3 = \pm\sqrt{d_{0,3}^2 - x_3^2 - y_3^2}$$

[138] 수학식 5 내지 수학식 7에서, (x_3,y_3,z_3)는 제3 앵커 노드의 좌표 값들이다. 제3 앵커 노드의 좌표 값들은 2개가 존재할 수 있다. 이들 좌표 값들 중 하나는 제5 단계에서 최종적으로 선택될 수 있다.

[139] 제5 단계에서는 도 9에 도시한 바와 같이 임의의 노드를 선택한다. 임의의 노드는 오더러 및 제1 내지 제3 앵커 노드들과 RTT 값들을 측정하기 위한 것이다. 임의의 노드는 노드#4(Node #4) 또는 제4 노드로 지칭될 수 있고, 그 좌표는 (x_4,y_4,z_4)로 표현될 수 있다.

[140] 임의의 노드를 이용하면, 오더러를 기준으로 제1 앵커 노드, 제2 앵커 노드 및 제3 앵커 노드의 좌표들을 확정지은 상태에서 다음의 수학식 8 내지 수학식 11을 통해 임의의 다른 참여 노드의 좌표 값의 존재를 보장할 수 있다.

[141] [수식8]

$$\sqrt{x_4^2 + y_4^2 + z_4^2} = d_{0,4}$$

[142] [수식9]

$$\sqrt{(x_4 - d_{0,1})^2 + y_4^2 + z_4^2} = d_{1,4}$$

[143] [수식10]

$$\sqrt{(x_4 - x_2)^2 + (y_4 \pm y_2)^2 + z_4^2} = d_{2,4}$$

[144] [수식11]

$$\sqrt{(x_4 - x_3)^2 + (y_4 - y_3)^2 + (z_4 \pm z_3)^2} = d_{3,4}$$

[145] 수학식 8 내지 수학식 11에서 (x_4,y_4,z_4)는 임의의 참여 노드들 각각의 좌표 값들에 대응될 수 있다.

[146] 이와 같이, 오더러와 3개의 앵커 노드들을 통해 오더러를 원점으로 하는 3차원 좌표계를 생성하고, 생성된 3차원 좌표계에서 임의의 참여 노드의 좌표를 계산할 수 있다.

[147] 한편, 전술한 제6 단계를 좀더 구체적으로 설명하면, 제6 단계에서는 3차원 좌표계가 생성된 후에, 모든 참여 노드들 각각이 오더러, 앵커 노드들 및 임의의 참조 노드들로 요청 메시지를 보내도록 구성될 수 있다. 여기서 오더러의 좌표 값과, 앵커 노드들의 좌표 값들과 임의의 참조 노드들의 좌표 값들은 검증 노드나 오더러에서 식별가능하다. 즉, 위의 제5 단계와 유사하게, 도 10에 도시한 바와 같이, 임의의 참여 노드의 좌표를 구할 수 있다. 임의의 참여 노드는 노드#k(Node #k) 또는 제k 노드로 지칭될 수 있고, 그 좌표는 (x_k,y_k,z_k)로 표현될 수 있다.

[148] 또 한편으로, 분산 환경에서는 좌표 값을 식별할 수 있는 참여 노드에 대한

정보를 얻기가 쉽지 않다. 즉, 위에서 설명한 제1 단계 내지 제5 단계의 프로세스에서 예컨대 역행렬이 행렬 형태로 존재하지 아니하는 미확인 케이스(case)가 발생할 수 있다. 이러한 문제 발생을 대비하기 위해, 본 실시예에서는 새로운 참여 노드가 앵커 노드들을 포함하고 충분한 개수로 임의로 선택되는 노드들과 접촉하도록 구성될 수 있다. 또한, 새로운 참여 노드를 충분한 개수의 임의로 선택되는 노드들과 접촉시키는 경우, 해당 노드의 좌표가 중복 결정되는 케이스가 유발될 수 있으므로, 이러한 문제 발생을 고려하여 본 실시예에서는 근사해를 구하여 그 오차를 최소화할 수 있다.

[149] 또한, 전술한 제7 단계를 좀더 구체적으로 설명하면 다음과 같다. 먼저 참여 노드의 좌표 값은 시간에 따라 변하는 네트워크 조건으로 인해 변경될 수 있다. 그 경우, 좌표 값을 효율적으로 업데이트하기 위해 자체 교차 검사 알고리즘을 수행한다.

[150] 자체 교차 검사 알고리즘은 도 11에 도시된 바와 같이 오더러가 이미 그들의 좌표 값을 식별하고 있는 참여 노드 세트에 요청 메시지를 전송하고, 오더러가 제6 단계에서와 같은 방법으로 응답 메시지를 기반으로 참여 노드들의 대략적인 좌표값들을 계산할 수 있다.

[151] 즉, 제1 노드(Node #1)의 좌표 (x_1,y_1,z_1), 제2 노드(Node #2)의 좌표 (x_2,y_2,z_2), 제3 노드(Node #3)의 좌표 (x_3,y_3,z_3), 제4 노드(Node #4)의 좌표 (x_4,y_4,z_4) 및 임의의 제N번째 노드(Node #N)의 좌표 (x_N,y_N,z_N)를 구하고, 일정 시간 후에 좌표값을 식별하고 있는 각 노드와 오더러인 제0 노드(Node #0)의 좌표 (x_0,y_0,z_0)와의 거리를 통해 오더러의 좌표를 다시 구할 수 있다.

[152] 그리고, 참여 노드들의 거리와 기설정 좌표 값들에 의해 얻어지는 오더러의 좌표가 0이 아닌(non-zero) 경우, 좌표계를 새롭게 업데이트한다. 특히, 좌표계의 업데이트에 따른 오버헤드를 줄이기 위해, 본 실시예에 따른 블록 전송 시스템은 오더러의 좌표 변동 크기가 임계값보다 큰 경우에만 좌표계를 업데이트하도록 구성될 수 있다. 결과적으로 제어 메시지 복잡도는 $O(n_{nd})$이다.

[153] 전술한 지연 추정 프로세스를 이용하는 지능한 블록 전송 시스템은, 블록 트래픽이 급증함에 따라 업로드 채널을 통해 블록 데이터를 전달하기 위해 더 많은 네트워크 리소스가 필요할 수 있다. 즉, 때때로 추가 지연이 발생할 수 있고, 그 경우 사용가능한 네트워크 리소스 양이 제한되어 블록체인 서비스 품질(QoS)이 저하될 수 있다. 실제로 업로드 대역폭은 위에서 언급한 클라우드 환경의 다운로드 대역폭보다 업로드 대역폭이 제한될 수 있고, 이러한 환경에서 각 부모 노드가 블록의 여러 복사본을 해당 자식 노드로 전송해야 하기 때문에 블록체인 서비스에서 병목 현상이 발생할 수 있다. 이에 본 실시예에서는 블록체인 QoS 및 확장성을 향상시킬 수 있는 능력을 고려하여 P2P 네트워크 상에서 피어 스트레스를 유지할 수 있도록 구성된다.

[154] 즉, 본 실시예의 지능형 블록 전송 시스템에서는 중복 블록 전송 횟수를 제어하여 전체 평균 지연을 최소화하고, 피어 스트레스를 처리하기 위해 P2P

멀티캐스트 기술을 채택한다. P2P 멀티캐스트 기술에서는 다음의 문제를 고려할 수 있다. 즉,주어진 업로드 대역폭으로 얼마나 많은 자식 노드를 지원할 수 있는지, 참여하는 노드를 어떻게 클러스터링할 것인지, 그리고 하이브리드 P2P 멀티캐스트 트리를 어떻게 구성할 것인지를 고려할 수 있다.

[155] 도 12 내지 도 15는 도 2의 블록 전송 시스템에 채용할 수 있는 블록 전송을 위한 유효 대역폭 추정 프로세스를 설명하기 위한 도면들이다.

[156] P2P 멀티캐스트 기술에서, 먼저 지원 가능한 자식 노드의 수는 블록 트래픽의 버스트를 결정하는 블록 생성 정책과 밀접하게 결합되며 블록 트래픽을 전달하기 위한 유효 대역폭은 버스트(bursts)에 따라 달라진다.

[157] 먼저, 블록 생성 정책과 블록 트래픽에 대한 즉, 블록 전송(block transmission)을 위한 유효 대역폭 추정 프로세스(effective bandwidth estimating process)를 설명하면 다음과 같다.

[158] 유효 대역폭 추정 프로세스에서 블록은 소정 개수(

N_{trx}^{blk}

)의 트랜잭션들이 오더러에 도착하자마자 생성된다고 가정한다. 또한, 인접한 두 트랜잭션들 사이의 시간 간격은 통상 사용되는 것과 같이 기하급수적으로 분포한다고 가정한다.

[159] 중심 극한 정리(central limit theorem, CLT)에 의하면,

N_{trx}^{blk}

트랜잭션들이 도착할 때까지의 시간 간격은,

N_{trx}^{blk}

이 소정의 기준값 이상으로 클 때, 즉 다음의 수학식 12와 같이 표현될 수 있을 때, 도 12와 같은 가우시안(Gaussian) 모델(G)로 나타낼 수 있다.

[160] [수식12]

$$G\left(N_{trx}^{blk}/\lambda, \sqrt{N_{trx}^{blk}}/\lambda\right)$$

[161] 여기서,

λ

는 지수 분포 확률 모델(exponentially distributed stochastic model)의 트랜잭션 속도이다.

[162] 위의 가정을 토대로 하면, 블록 전송 시간(

$\tilde{t}_{BLK}$

)은 릴레이 노드들(relay nodes)에서 블록 버퍼링 지연(block buffering delay)을 가능한한 피하기 위해 다음 블록(the next block)이 생성되기 전에 주어진 성공 확률(given success probability)에서 이전 블록 전송(previous block transmission)을 완료하도록 결정될 수 있다. 즉, 블록 전송 시간

$\tilde{t}_{BLK}$

는 다음의 수학식 13과 같이 구해진다.

[163] [수식13]

$$\left(N_{trx}^{blk} - k \cdot \sqrt{N_{trx}^{blk}}\right) / \lambda$$

[164] 여기서,

k

는 가중치로서 도 12에 도시한 바와 같이 미리 결정된 성공 확률(

P_{fin}

)을 고려하면서 고정될 수 있다. 그리고

P_{fin}

은 이전 블록이 다음 블록이 생성되기 전에 이미 전송되는 확률을 나타낸다.

P_{fin}

이 증가함에 따라 즉,

$\tilde{t}_{BLK}$

이 더 작은 숫자로 설정될 때, 업로드 대역폭 사용을 희생하면서 더 작은 블록 버퍼링 시간이 확률적으로 보장되며, 그 반대의 경우도 마찬가지이다.

[165] 위에서 언급한 조건에 따라 유효 대역폭은 다음의 수학식 14와 같이 추정될 수 있다.

[166] [수식14]

$$BW_{eff} = \frac{N_{trx}^{blk} \cdot B_{TX} + B_{blk\_hd}}{\tilde{t}_{BLK}}$$

[167] 여기서

B_{TX}

는 트랜잭션 크기이고,

B_{blk_hd}

는 블록 헤더 크기이다. 이론적 유효 대역폭은

λ

,

N_{trx}^{blk}

및

P_{fin}

의 측면에서 도 13 내지 도 15와 같이 예시될 수 있다.

[168] 도 13 내지 도 15를 참조하면,

N_{trx}^{blk}

이 클 때, 유효 대역폭은

λ

에 따라 선형적으로 증가한다는 것이 명백하게 관찰된다. 반면에

λ

가 상대적으로 크면, 유효 대역폭은

N_{trx}^{blk}

에 대해 거의 기하급수적으로 감소하고,

P_{fin}

이 커질수록 감소율은 더 민감해진다.

[169] 일례로, 도 13은

P_{fin}

이 84.13%일 때, 도 14는

P_{fin}

이 93.32%일 때, 그리고 도 15는

P_{fin}

이 97.72%일 때의

λ

,

N_{trx}^{blk}

및

P_{fin}

측면에서의 이론적인 유효 대역폭을 각각 나타낸다.

[170] N_{trx}^{blk}

트랜잭션들이 도착할 때까지의 시간 간격은, 때때로 낮은 확률로, 너무 커서 블록체인의 서비스 품질(QoS)을 심각하게 저하시킬 수 있다. 이러한 원하지 않는 경우를 방지하기 위해, 가장 오래된 트랜잭션의 버퍼링 시간이 임계값

$T_{blk}^{\max}$

(도 12 참조)보다 더 크면, 도달하는 트랜잭션들의 개수가

N_{trx}^{blk}

보다 작더라도, 즉시 블록을 생성하도록 구성될 수 있다.

[171] 다음은 가우시안 혼합 모델(gaussian mixture mdel, GMM) 기반 노드 클러스터링(clustering) 및 클러스터 리더 선택 프로세스(cluster leader selecting

process)를 설명하기로 한다.

[172] 클러스터링 프로세스에서는 미리 구해진 참여 노드들의 3D 좌표 값을 기반으로 여러 클러스터로 그룹화하고 이를 토대로 참여 노드들의 하이브리드 P2P 멀티캐스트 트리를 구성할 수 있다. 본 실시예에서는 기계학습 분야 등에 잘 알려져 있는 GMM 기반 클러스터링 알고리즘을 사용하나 이에 한정되지는 아니하며, GMM 기반 클러스터링과 동일하거나 유사한 기능을 수행하는 기존의 다른 클러스터링 알고리즘을 사용할 수 있음은 물론이다.

[173] GMM은 유한한 집합의 가우시안 확률 밀도 함수를 사용하여 샘플 분포를 추정한다. 기본적으로 GMM 기반 클러스터링 알고리즘은 데이터 포인트의 확률 밀도를 최대화하도록 설계된다. 즉, 각 클러스터는 가우시안 확률 밀도 함수로 표현되고 각 포인트는 최대 확률 밀도로 클러스터에 삽입된다. 본 실시예의 블록 전송 시스템에서는 GMM 파라미터 추정에 널리 사용되는 기댓값 최대화(expectation maximum, EM) 알고리즘을 자체 구현하여 적용하였다. GMM에서 EM 알고리즘의 복잡성은

$O(N \cdot K \cdot D^3)$

이다.

[174] 여기서, N은 데이터의 개수이고, K는 가우시안 성분들의 개수이고, D는 데이터 포인트의 차원이다.

[175] 한편, GMM 기반 클러스터링 알고리즘은 참여 노드의 좌표가 3차원에 불과하기 때문에 블록 전송 시스템에서 합리적인 복잡성을 갖고 실행될 수 있다.

[176] 클러스터링이 완료되면, 지연 추정 프로세스를 통해 구한 지연 및 유효 대역폭 추정 프로세스를 통해 구한 업링크 대역폭을 고려하여 소정의 노드를 각 클러스터의 클러스터 리더로 선택할 수 있다. 이와 같이, 클러스터 #i(cluster #i)에서의 노드 #j(node #j)에 대한 측정은 다음의 수학식 15와 같이 정의될 수 있다.

[177] [수식15]

$$S_j\left(d_{0,j}, C_j^{\max}\right) = (1-\omega)\frac{d_{0,j}}{\max\limits_{p\in P_i}\left\{d_{0,p}\right\}} + \omega\frac{\max\limits_{p\in P_i}\left\{C_p^{\max}\right\} - C_j^{\max}}{\max\limits_{p\in P_i}\left\{C_p^{\max}\right\}}$$

[178] 여기서

ω

는 실수,

P_i

는 클러스터 #i의 노드 집합,

$C_j^{\max}$

는 지원 가능한 자식 노드의 수를 각각 나타낸다. 업로드 대역폭은 유효 대역폭

추정 프로세스에서 추정된 유효 대역폭으로 나누어 계산된다.

[179] 본 실시예의 블록 전송 시스템에서는 측정값이 가장 작은 노드를 클러스터 리더로 선택할 수 있다.

[180] 한편, GMM 기반 클러스터링 알고리즘은 최적의 클러스터 개수를 자동으로 제공하지 않는 것으로 잘 알려져 있다. 이에 본 실시예에서는 위의 클러스터링과 클러스터 리더 선택 과정을 후술하는 하이브리드 P2P 멀티캐스트 트리 구성 과정과 반복적으로 수행하여 최적의 클러스터 수와 클러스터 리더를 결정할 수 있다.

[181] 다음은 하이브리드 P2P 멀티캐스트 트리 구성(hybrid P2P multicast tree constructing) 프로세스를 설명하기로 한다.

[182] 본 프로세스에 대한 설명에 있어서, 참여 노드, 클러스터 간의 모든 지연 값과 각 노드에서 지원 가능한 자식 노드의 수가 제공된다는 가정 하에, 최소 평균 지연을 갖는 하이브리드 P2P 멀티캐스트 트리를 구성하기 위한 수정된 다익스트라 알고리즘(modified Dijkstra's algorithm, MDA)을 이용할 수 있다. MDA는 본 실시예의 블록 전송 시스템의 구성 요소로 채택될 수 있다. 블록 전송 시스템에서는 지연 추정 프로세스를 통해 얻은 지연 정보를 기반으로 평균 지연을 최소화하기 위해 클러스터 수와 MDA 기반 하이브리드 P2P 멀티캐스트 트리를 반복적으로 결정할 수 있다. 그리고 클러스터 리더에서 발생되는 지연은 다음의 수학식 16에 의해 계산될 수 있다.

[183] [수식16]

$$d_{0,p_i^{leader}} = \begin{cases} d_{0,p_i^{leader}} & \text{if directly received from the orderer} \\ d_{0,p_1^{leader}} + \cdots + d_{p_{k-1}^{leader},p_k^{leader}} + d_{p_k^{leader},p_i^{leader}} & \text{otherwise} \end{cases}$$

[184] 여기서,

$d_{0,p_i^{leader}}$

는 오더러로부터 직접 받은 경우(if directly received from the orderer)의 오더러와 클러스터 리더(

p_i^{leader}

) 간의 지연이고,

$d_{p_{k-1}^{leader},p_k^{leader}}$

는 오더러로부터 직접 받은 경우가 아닌(otherwise), 두 클러스터 리더들(

p_{k-1}^{leader}

및

p_k^{leader}

) 간의 지연이다.

[185] 본 실시예의 멀티캐스트 트리를 통해 임의의 블록체인 노드가 경험하는 지연은 다음의 수학식 17에 의해 추정될 수 있다.

[186] [수식17]

$$d_{0,j} \cong d_{0,p_i^{leader}} + d_{p_i^{leader},j}$$

[187] 여기서,

$d_{p_i^{leader},j}$

는 i-번째 클러스터의 j-번째 노드와 i-번째 클러스터 내 클러스터 리더(p_i^{leader}) 간의 지연이다.

[188] 전술한 추정 결과에 의하면, 멀티캐스트 트리를 통해 참여 노드들의 전체 지연 합은 다음의 수학식 18과 같이 표현될 수 있다.

[189] [수식18]

$$\sum_{i=1}^{n^{cluster}} d_{0,p_i^{leader}} + \sum_{i=1}^{n^{cluster}} \sum_{j=1}^{m_i} \left(d_{0,p_i^{leader}} + d_{p_i^{leader},j} \right)$$

[190] 여기서

$n^{cluster}$

는 클러스터의 수이고,

m_i

는 i-번째 클러스터에 참여하는 노드의 개수이다. 따라서 평균 지연은 수학식 19로 표현될 수 있다. 수학식 19는 오더러에 의해 처리되는 자식 노드들의 개수가 오더러에 의해 지원가능한 자식 노드들의 개수 이하이고, j-번째 노드에 의해 처리되는 자식 노드들의 개수가 j-번째 노드에 의해 지원가능한 자식 노드들의 개수 이하에 대하여 하기 조건으로(subject to) 적용될 수 있다.

[191] [수식19]

$$\frac{1}{N_{pn}} \sum_{i=1}^{n^{cluster}} m_i \cdot d_{0,p_i^{leader}} + \frac{1}{N_{pn}} \sum_{i=1}^{n^{cluster}} \sum_{j=1}^{m_i} d_{p_i^{leader},j}$$

$$\text{subject to } c_0 \le C_0^{max} \text{ and } c_j \le C_j^{max},$$

[192] 여기서,

N_{pn}

은 멀티캐스트에 참여하는 참여 노드들의 개수이다. 참여 노드는 엔드 시스템(end system)을 포함할 수 있다. 참여 노드들의 개수는 수학식 20으로 표현될 수 있다.

[193] [수식20]

$$\left(N_{pn} = \sum_{i=1}^{n^{cluster}} m_i \right)$$

[194] 또한, 수학식 19에서,

c_0

는 오더러에 의해 처리되는 자식 노드들의 개수를,

c_j

는 j-번째 노드에 의해 처리되는 자식 노드들의 개수를 각각 나타내고,

$C_0^{\max}$

는 오더러에 의해 지원가능한 자식 노드들의 개수를,

$C_j^{\max}$

는 j-번째 노드에 의해 지원가능한 자식 노드들의 개수를 각각 나타낸다.

[195] 그리고 수학식 19에서, 오더러와 j-번째 노드의 처리되는 자식 노드들의 개수가 지원가능한 자식 노드들의 최대 개수 이하인 조건은 스트레스(stress) 및 정도 제한(degree constraint)과 관련이 있습니다. 위에서 언급한 바와 같이, 수정된 다익스트라 알고리즘은 위에서 언급한 문제에 대한 효과적인 솔루션을 제공할 수 있음을 보여준다.

[196] 이와 같이 수정된 다익스트라 알고리즘을 구현하여 평균 지연을 최소화하는 하이브리드 P2P 멀티캐스트 트리를 구성할 수 있다.

[197] 수정된 다익스트라 알고리즘을 이용하여 평균 지연을 최소화하는 하이브리드 P2P 멀티캐스트 트리를 구성하는 간략히 예시하면 다음과 같다.

[198] 먼저, 오더러(O)를 공집합(S)에 추가한다. 그리고 모든 참여 노드들 중 v-번째 클러스터에 속한 제1 노드(v)에 대하여, 특정한 제1 노드가 오더러에 인접하게 위치할 때, 오더러로부터 최소 가중 RTT를 갖는 특정 노드까지의 링크 비용

$D(v)$

를, 오더러로부터 제1 노드까지의 거리를 v-번째 클러스터에 참여하는 노드들의 개수로 나눈 값(

$d_{0,v}/m_v$

)으로 설정하고, 그렇지 않으면 무한대(infinity)로 설정한다.

[199] 다음, 오더러로부터 해당 노드까지의 링크 비용이 가장 작은 조건을 만족하고, 집합(S)에 속하지 않는 제2 노드(w)를 찾는다.

[200] [수식21]

$$d_{k,w} > ((m_k + m_w)/(m_k - m_w))\left(d_{v,p_k^{leader}} - d_{v,p_w^{leader}}\right)$$

[201] 임의의 클러스터에 속한 임의의 노드(k)에 대하여 상기의 수학식 21을 만족하는 제2 노드(w)가 존재하는 경우, 제1 노드(v)가 새로운 자식 노드들을 지원할 수 있으면, 제2 노드(w)를 집합(S)에 추가한다. 한편, 제1 노드(v)가 새로운 자식 노드들을 지원할 수 없으면, 제2 노드까지의 링크 비용

$D(w)$

를 상대적으로 큰 수 또는 무한대로 대체한 후, 또 다른 제2 노드를 찾는 단계로 되돌아간다.

[202] 그리고, 임의의 노드(k)에 대하여 상기의 수학식 21을 만족하는 제2 노드(w)가 존재하지 않는 경우, 하기의 수학식 22를 만족하는 임의의 노드(k)를 집합(S)에 추가하고, 제1 노드(v)가 새로운 자식 노드들을 지원할 수 있을 때, 제2 노드(w) 또는 임의의 노드(k)에 인접하고 집합(S)에 속하지 않은 모든 노드에 대해 링크 비용

$D(v)$

를 업데이트한다. 이를 나타내면 수학식 22와 같다.

[203] [수식22]

$$D(v) = \min\left\{D(v), \frac{d_{0,w} + d_{w,v}}{m_v}\right\}$$

[204] 마지막으로, 모든 노드들이 집합(S) 내에 있으면, 현재의 프로세스를 종료하고, 그렇지 않으면, 또 다른 제2 노드를 찾는 단계로 되돌아간다.

[205] 다음은 트리 정보에 대한 제어 메시지 포맷 및 전달 프로세스를 설명하기로 한다.

[206] 제어 메시지 오버헤드는 참여 노드의 수가 증가함에 따라 P2P 백본 네트워크에 큰 부담이 될 수 있다. 이에 본 실시예에서는 효과적인 제어 메시지 포맷과 이를 이용한 전달 프로세스를 주요 특징으로 포함한다.

[207] 도 16은 도 2의 블록 전송 시스템에 채용할 수 있는 트리 정보 전달을 위한 제어 메시지 포맷에 대한 예시도이다. 그리고 도 17 및 도 18은 도 16의 제어 메시지를 이용한 트리 구조 식별 과정을 설명하기 위한 예시도들이다.

[208] 도 16을 참조하면, 제어 메시지 포맷(180)은 하이브리드 P2P 멀티캐스트 트리를 통해 전달될 때 오버헤드를 줄이기 위해 계층적으로 설계된다. 즉, 제어 메시지 포맷(180)은 헤더(181)와 페이로드를 포함하고, 헤더(181)는 버전(version), 메시지 유형(message type), 메시지 길이(message length)에 대한 정보를 담고, 페이로드는 소정의 순번들(#0 내지 #n)에 따라 복수의 노드들의 트리 정보를 각각 순차적으로 나열한 형태를 구비할 수 있다.헤더(181)와 각 노드의 트리 정보는 32 바이트(byte)의 길이를 가질 수 있으나, 이에 한정되지는 않는다.

[209] 하나의 노드의 트리 정보(도 18의 183 참조)는 노드 IP(Node IP), 자식 노드 수(the number of child nodes) 및 자식 노드 목록의 오프셋(offset of child node lists)을 각각 표현하는 필드들을 구비한다(도 18의 184, 185 및 186 참조).

[210] 여기서 노드 IP(Node IP)는 현재 노드의 IP(internet protocol) 주소를 나타내고, 자식 노드 수(the number of child nodes)는 하이브리드 P2P 멀티캐스트 트리를 통해 현재 노드에 연결된 자식 노드의 수를 나타내며, 자식 노드 목록의 오프셋(offset of child node lists)은 제어 메시지에서 첫 번째 자식 노드의 위치를

나타낸다.

[211] 전술한 제어 메시지 포맷의 적용과 그 전달 프로세서를 예시하면 다음과 같다. 도 17 및 도 18에 예시한 바와 같이, 0 내지 13으로 각각 표시된 노드 #0 내지 노드 #13이 부모 노드와 자식 노드의 관계로 직간접으로 연결되어 있을 때, 제어 메시지가 노드 #0(170)로부터 노드 #2(172)에 도착하면, 노드 #2(172)는 제어 메시지의 자식 노드 수 필드(185)를 통해 자식 노드의 수를 파악하고, 자식 노드 목록의 오프셋 필드(186)의 바이트 값 예컨대 384 바이트가 가리키는 첫번째 자식 노드의 위치로부터 노드 #5 및 노드 #6가 자신의 자식 노드들임을 알 수 있다.

[212] 또한, 노드 #2(172)는 제어 메시지의 하위 트리에 대한 불필요한 정보를 제거하여 제어 메시지를 재구성할 수 있다. 불필요한 정보는 노드 #1(171)과 그 하위 노드들(#3, #4, #7 내지 #9)에 대한 정보를 지칭할 수 있다. 재구성된 제어 메시지는 해당 자식 노드들 즉, 노드 #5와 노드 #6으로 전달된다.

[213] 이와 같이 제어 메시지의 크기는 하이브리드 P2P 멀티캐스트 트리를 통해 이동함에 따라 명백하게 줄어들 수 있다. 더욱이, 제어 메시지가 자식 노드가 없는 리프 노드들에 도착하는 즉시 모든 참여 노드들 사이에 하이브리드 P2P 멀티캐스트 트리를 완전히 파악할 수 있게 된다.

[214] 도 19는 도 1의 블록 전송 시스템이 적용될 수 있는 멀티 클라우드의 네트워크 토폴로지에 대한 예시도이다. 도 20은 도 19의 멀티 클라우드 네트워크에 채용할 수 있는 멀티 클라우드 테스트베드의 시스템 아키텍처에 대한 예시도이다.

[215] 본 실시예에 따른 블록 전송 시스템은 도 19 및 도 20과 같이 현실적인 멀티 클라우드 에뮬레이션 환경에서 그 성능을 검증할 수 있다.

[216] 멀티 클라우드 네트워크 토폴로지는, 도 19에 도시한 바와 같이, 클라우드 A(cloud A) 내지 클라우드 G(cloud G)의 복수의 클라우드들과, 특정 클라우드 예컨대 클라우드 A에 포함된 검증노드(100) 및 오더러(orderer, 300)와, 각 클라우드에 포함된 참여 노드들(500)을 포함한다. 각 참여 노드에는 소정의 일련 번호(1 내지 59) 중 하나가 부여될 수 있고, 오더러(300)에는 일련 번호 0이 부여될 수 있다. 클라우드들을 서로 적어도 하나 이상의 게이트웨이를 통해 서로 연결되고, 각 클라우드 내부의 노드들도 게이트웨이를 통해 서로 연결될 수 있다.

[217] 멀티 클라우드의 네트워크 토폴로지에서, 지연 값은 인터넷을 통해 관찰된 RTT 값을 고려하여 설정될 수 있다. 지연 값은 3㎳, 5㎳, 6㎳, 10㎳, 12㎳ 등으로 설정되었다.

[218] 테스트베드는 하이퍼레저 패브릭(hyperledger fabric), Docker(https://www.docker.com), Containernet(https://containernet.github.io), Mininet(http://mininet.org), C/C++, 파이썬(Python) 등과 같은 오픈 소스를 사용하여 구현될 수 있다. 멀티 클라우드 테스트베드의 시스템 아키텍처(200)는 도 20에 나타낸 바와 같이, 미니넷(Mininet)과 컨테이너넷(Containernet)을

사용하여 멀티 클라우드를 현실적으로 에뮬레이트할 수 있다.

[219] 멀티 클라우드 테이트베드의 시스템 아키텍처는 블록 생성 노드인 오더러(orderer, 300), 1개의 검증 노드, 그리고 59개의 참여 노드들이 있다고 가정하였다. 오더러를 참여 노드에 포함시키는 경우, 참여 노드들은 60개로 지칭될 수 있다.

[220] 멀티 클라우드 테이트베드의 시스템 아키텍처에서 노드 #0에 대응하거나 노드 #0에 탑재되는 오더러(orderer, 300)는 모듈형 아키텍처 플랫폼인 하이퍼레저 패브릭(200)를 포함하거나 이와 결합될 수 있다. 오더러(300)는 노드 #1(node #1), 노드 #6(node #6), 노드 #55(node #55)에 연결된다. 노드 #1, 노드 #6 및 노드 #55는 앵커 노드들에 대응된다.

[221] 예를 들어, 노드 #1(node #1)은 노드 #2 내지 노드 #5, 노드 #7, 노드 #9 및 노드 #10에 각각 연결되고, 노드 #9 및 노드 #10은 노드 #8에 공통 연결되고, 노드 #7은 노드 #13, 내지 노드 #15에 각각 연결된다. 또한 노드 #6(node #6)은 노드 #11(node #11)에 연결되고, 노드 #11은 노드 #12(node #12)에 연결된다.

[222] 실험 시, 블록의 헤더 크기와 트랜잭션 크기는 각각 1000 바이트와 4000 바이트로 설정하고, 하나의 블록의 트랜잭션 개수(N_{trx}^{blk})는 10, 15, 20, 25로 각각 설정하였다. 또한, 유효 대역폭 추정 프로세스의 k 값은 97.72%의 확률로 이전 블록 전송을 완료하기 위해 3로 고정되었다. 이는 매우 보수적이어서 결과 데이터가 성능의 하한에 가깝게 나타났다. 즉, 최대 블록 전송 시간 $T_{blk}^{\max}$ 을 수학식 23과 같이 설정할 수 있다.

[223] [수식23]

$$\left(N_{trx}^{blk}+3\cdot\sqrt{N_{trx}^{blk}}\right)/\lambda$$

[224] 도 21은 도 2의 블록 전송 시스템에서의 지연 추정 프로세스에 대한 성능 검증 결과를 나타낸 그래프이다. 도 22a 내지 도 22c는 도 21의 지연 추정 프로세스와 관련하여 노드 #0을 기준으로 모든 노드들 각각에 대해 측정한 지연(measured delay)과 각 노드의 좌표 값을 이용하여 추정된 지연(estimated delay using coordinate values)을 노드별(Node ID)로 비교하여 나타낸 그래프들이다.

[225] 본 실험 결과에서는 블록 전송 시스템의 지연 추정 프로세스의 성능을 보여준다. 본 실험은 도 19를 참조하여 앞서 설명한 것과 같은 네트워크 토폴로지를 통해 실행되었다. 모든 참여 노드들 중 노드 #1, 노드 #2, 노드 #3이 연이어 앵커 노드로 선택되고 3차원(3D) 공간에서 결과적인 노드 위치들이 원형 점 형태로 도 21에 도시되어 있다.

[226] 도 21을 참조하면, 노드들이 3차원 공간에 넓게 분포되어 있음을 분명히 알 수 있으며, 이는 노드들이 2차원 평면에서 표현될 경우 큰 지연 추정 오차가 불가피함을 의미한다.

[227] 지연 추정 프로세스의 성능은 도 22a 내지 도 22c에 요약되어 있다. 한 쌍의 노드들 사이의 지연 값을 모두 측정하고 앞서 언급한 좌표 값들 간의 유클리드 거리(Euclidean distance)를 계산하여 해당 지연 값을 추정하였다. 성능 검증 결과, 평균 오류(average error)는 약 0.82ms이므로 추정 오류가 허용가능함을 알 수 있다.

[228] 도 23 및 도 24는 도 2의 블록 전송 시스템에서의 클러스터링 및 하이브리드 P2P 멀티캐스트 트리 컨스트럭팅 프로세스(clustering & hybrid P2P multicast tree constructing processes, CCP)에 대한 성능 검증 결과를 나타낸 그래프들이다.

[229] 본 성능 검증 과정에서는 클러스터의 수를 2에서 10으로 변경하여 도 6 내지 도 9를 참조하여 설명한 지연 추정 프로세스를 실행 과정에서 얻어지는 좌표 값들로 본 프로세스를 실행할 수 있다.

[230] 성능 검증 과정 중 클러스터 리더 선택은 경험적으로 0.5로 고정하고, 트랜잭션 속도(

λ

)는 15와 30으로 설정하고, 모든 참여 노드의 상향링크 대역폭 값은 10Mbps로 동일하게 설정하였다. 트랜잭션 속도(

λ

)가 15일 때 유효 대역폭은 1,480kbps이고 지원 가능한 자식 노드의 수는 6이다.

[231] 클러스터의 개수에 따른 평균 지연과 최대 지연(average delay and maximum delay according to the number of clusters) 예컨대, 클러스터의 개수를 2개에서 10개로 변경했을 때의 평균 지연(avg. delay) 값과 최대 지연(max. delay) 값을 나타내면 표 1과 같다.

[232] [표1]

	λ= 15		λ= 30	
클러스터수	Avg. delay (ms)	Max. delay (ms)	Avg. delay (ms)	Max. delay (ms)
2	15.55	29.80	22.59	31.05
3	15.63	29.80	17.42	31.05
4	15.64	29.80	**16.02**	31.05
5	15.55	28.85	17.07	28.85
6	**15.54**	28.84	17.00	28.84
7	15.66	28.84	19.10	28.84
8	15.89	29.64	41.16	62.85
9	16.38	33.30	16.78	31.02
10	16.49	33.30	28.62	56.55

[233] 표 1에 나타낸 바와 같이, 트랜잭션 속도(λ)가 15일 때 그리고 클러스터 개수가 6개일 때 평균 지연값이 최소화됨을 알 수 있다. 그 결과 멀티캐스트 트리는 도 23에 도시한 바와 같다.또한, 트랜잭션 속도(λ)가 30일 때 유효 대역폭과 해당 자식 노드의 수는 각각 2,960kbps와 3이며, 클러스터 수가 4일 때 평균 지연이 최소화되는 것을 확인할 수 있다.

[234] 한편, 본 실시예의 블록 전송 시스템은 평균 지연만을 최소화하도록 설계되어 있기 때문에 최대 지연은 도 24에 나타낸 바와 같이 때때로 약간 증가할 수 있다. 도 23 및 도 24의 두 트리들의 차이가 가시적으로 확연히 드러나는 것을 알 수 있다.

[235] 도 25 및 도 26은 도 23 및 도 24의 CCP와 관련하여 업링크 대역폭이 200Mbps일 때의 평균 블록 도착 시간과 최대 블록 도착 시간의 지연 비교를 위한 그래프들이다.

[236] 본 성능 검증에서는 본 실시예(proposed)의 블록 전송 시스템을 가십(Gossip), 확장형 가십(Enhanced Gossip), 유니캐스트(Unicast)와 비교한다. 가십(Gossip)은 하이퍼레저 패브릭(hyperledger fabric) 플랫폼의 기본 임베디드 전송 알고리즘 중 하나이며, 확장형 가십(Enhanced Gossip)은 블록을 이웃 노드로 효과적으로 전달하기 위해 TTL(time-to-live) 값을 유지하기 위한 카운터 필드를 추가로 포함하며, 유니캐스트(Unicast)는 오더러가 모든 참여 노드들에 개별적으로 블록을 보내도록 구성된다.

[237] 본 성능 검증에서는 25초 동안 실험을 수행하고 12초에 트랜잭션 속도(λ)를 15에서 30으로 변경한다. 실험 중 업링크 대역폭 값은 50Mbps, 100Mbps 및

N_{trx}^{blk})는 10, 15, 20, 25로 각각 설정하였다. 전체 실험 결과는 도 25 및 도 26과 및 표 2에 요약되어 있다.

[238] [표2]

업링크 대역폭 (Mbps)	알고리즘	평균 블록 도착 시간(ms)				최대 블록 도착 시간(ms)			
		N_{trx}^{blk} = 10	N_{trx}^{blk} = 15	N_{trx}^{blk} = 20	N_{trx}^{blk} = 25	N_{trx}^{blk} = 10	N_{trx}^{blk} = 15	N_{trx}^{blk} = 20	N_{trx}^{blk} = 25
50	Gossip	2	8	4	3	651	478	466	632
	Enhanced Gossip	156	142	180	183	775	796	896	855
50	Unicast	226	363	931	1,386	509	916	2,617	5,335
	Proposed	55	76	117	141	119	129	197	250
100	Gossip	107	117	132	143	1,455	1,475	1,503	1,691
	Enhanced Gossip	110	139	136	149	725	785	807	1,005
	Unicast	127	185	249	316	252	377	576	823
	Proposed	45	58	71	86	84	107	132	192
200	Gossip	101	103	121	128	1,464	1,462	1,504	1,576
	Enhanced Gossip	86	99	119	125	714	735	800	929
	Unicast	81	113	148	181	163	228	296	374
	Proposed	31	40	52	61	68	91	128	157

[239] 표 2에 나타낸 바와 같이, 가십(Gossip)은 노드들의 집합을 무작위로 선택하여 그들에게 블록을 전달하는 방식으로 구현되기 때문에 도 26과 같이 블록 도달 시간의 최대값이 가장 크다. 확장형 가십(Enhanced Gossip)은 가십의 성능을 약간 향상시키고 있다. 그리고 유니캐스트(Unicast)의 경우 오더러의 액세스 네트워크 대역폭이 병목 현상을 유발할 수 있다.

[240] 도 25 및 도 26과 표 2에서 블록 도달 시간의 평균값과 최대값은 오더러의 액세스 네트워크 대역폭의 제한으로 인해 크게 증가함을 명백히 알 수 있다. 특히, 이러한 값은 트랜잭션 속도(λ)가 증가함에 따라 급격히 증가한다. 즉, 업링크 대역폭이 50Mbps이고 블록의 트랜잭션 개수(N_{trx}^{blk})가 20일 때 평균 및 최대 블록 도달 시간은, λ=15(0~12s)의 경우에 477ms 및 894ms에서 변경되었고, λ=30(12~12초)의 경우에 1,144ms 및 2,617ms로 변경되었다. 반면에, 본 실시예(Proposed)의 블록 전송 시스템은 비교예들의 다른 어떤 알고리즘보다

블록 도달 시간의 평균값과 최대값을 훨씬 더 작게 지원할 수 있음을 분명히 알 수 있다. 위와 같은 결과의 이유는 본 실시예의 시스템이 추정된 지연 값을 기반으로 트래픽 부하를 여러 참여 노드에 체계적으로 할당할 수 있기 때문이다.

[241] 한편, 본 실시예의 경우 N_{trx}^{blk}에 따라 유효 대역폭이 감소함에도 불구하고 블록 도달 시간의 평균과 최대값은 약간 증가함을 알 수 있다. 실제로, 이러한 현상은 N_{trx}^{blk}에 따라 블록 크기가 증가하기 때문에 릴레이 노드들에서 누적된 전송 지연에 의해 발생할 수 있다.

[242] 전술한 성능 검증 결과에 의하면, 본 실시예의 블록 전송 시스템은 블록체인 확장성과 트랜잭션 처리량을 눈에 띄게 향상시킬 수 있음을 알 수 있다.

[243] 도 27은 본 발명의 다른 실시예에 따른 블록 전송 시스템의 구성에 대한 개략적인 블록도이다.

[244] 도 27을 참조하면, 블록 전송 시스템(1000)은 검증 노드, 오더러 또는 참여 노드 중 적어도 어느 하나의 노드에 설치되거나 결합될 수 있으며, 그 역도 가능하다. 이러한 블록 전송 시스템(1000)은 적어도 하나의 프로세서(1100) 및 메모리(1200)를 포함할 수 있다. 또한, 블록 전송 시스템(1000)은 네트워크와 연결되어 통신을 수행하는 송수신 장치(1300)를 더 포함할 수 있다. 또한, 블록 전송 시스템(1000)은 입력 인터페이스 장치(1400), 출력 인터페이스 장치(1500), 저장 장치(1600) 등을 더 포함할 수 있다. 블록 전송 시스템(1000)에 포함된 각각의 구성 요소들은 버스(bus, 1700)에 의해 연결되어 서로 통신을 수행할 수 있다.

[245] 프로세서(1100)는 메모리(1200) 및 저장 장치(1600) 중에서 적어도 하나에 저장된 프로그램 명령(program command)을 실행할 수 있다. 프로세서(1100)는 중앙 처리 장치(central processing unit, CPU), 그래픽 처리 장치(graphics processing unit, GPU), 또는 본 발명의 실시예들에 따른 방법들이 수행되는 전용의 프로세서를 의미할 수 있다.

[246] 프로그램 명령(program command)은 본 실시예의 제어 메시지 포맷을 갖는 블록을 멀티태스크 트리 구조로 전송하는 블록 전송 시스템의 적어도 하나의 구성요소를 소프트웨어적으로 구성하기 위한 명령, 적어도 일부의 구성요소를 구동하기 위한 명령, 적어도 일부의 구성요소의 기능을 실행하기 위한 명령 등을 포함할 수 있다.

[247] 일례로, 프로그램 명령은, 네트워크 모니터링 및 관리를 위한 명령, 블록 생성을 위한 명령, 멀티태스크 트리 정보에 기초한 트리블록 전송을 위한 명령, 블록체인 인터페이스를 위한 명령, 네트워크 모니터링을 위한 명령, n-차원 좌표를 구성하는 명령, n-차원 좌표를 관리하는 명령, 업링크 대역폭을 모니터링하는 명령, 트랜잭션 속도를 모니터링하는 명령, 유효 대역폭을 계산하는 명령, 지원가능한 자식 노드들의 수를 계산하는 명령, 참여 노드들로부터 좌표 값을 수신하는 명령, 트랜잭션 속도 정보를 수신하는 명령,

모든 참여 노드들 사이의 엔드 투 엔드 지연을 추정하는 명령, 모든 참여 노드들의 클러스터링을 위한 명령, 클러스터 리더를 선택하기 위한 명령, 중첩 멀티캐스트 트리를 구성하는 명령, 네트워크 토폴로지 구성을 위한 명령, 본 실시예의 제어 메시지 포맷에 따라 제어 메시지를 생성하는 명령, 제어 메시지를 통해 트리 정보를 전달하는 명령, 메시지 전달 테이블을 업데이트하는 명령, 업데이트된 메시지 전달 테이블을 자식 노드로 전달하는 명령을 포함할 수 있다.

[248] 메모리(1200) 및 저장 장치(1600) 각각은 휘발성 저장 매체 및 비휘발성 저장 매체 중에서 적어도 하나로 구성될 수 있다. 예를 들어, 메모리(1200)는 읽기 전용 메모리(read only memory, ROM) 및 랜덤 액세스 메모리(random access memory, RAM) 중에서 적어도 하나로 구성될 수 있다.

[249] 송수신 장치(1300)는 근거리 무선 네트워크나 케이블 연결, 위성과의 통신, 범용 기지국과의 유선 또는 무선 통신, 모바일 에지 코아 네트워크나 코아 네트워크(core network)와의 아이디얼 백홀 링크(ideal backhaul link) 또는 넌(non)-아이디얼 백홀 링크의 연결 등을 위한 통신인터페이스나 서브통신시스템을 포함할 수 있다.

[250] 입력 인터페이스 장치(1400)는 키보드, 마이크, 터치패드, 터치스크린 등의 입력 수단들에서 선택되는 적어도 하나와 적어도 하나의 입력 수단을 통해 입력되는 신호를 기저장된 명령과 매핑하거나 처리하는 입력 신호 처리부를 포함할 수 있다.

[251] 출력 인터페이스 장치(1500)는 프로세서(1100)의 제어에 따라 출력되는 신호를 기저장된 신호 형태나 레벨로 매핑하거나 처리하는 출력 신호 처리부와, 출력 신호 처리부의 신호에 따라 진동, 빛 등의 형태로 신호나 정보를 출력하는 적어도 하나의 출력 수단을 포함할 수 있다. 적어도 하나의 출력 수단은 스피커, 디스플레이 장치, 프린터, 광 출력 장치, 진동 출력 장치 등의 출력 수단들에서 선택되는 적어도 하나를 포함할 수 있다.

[252] 이와 같이 본 실시예에서는 멀티 클라우드 상에서 확장 가능한 하이퍼레저 패브릭(hyperledger fabric) 블록체인을 위한 지능형 블록 전송 시스템을 제공할 수 있다. 지능형 블록 전송 시스템은 3차원 좌표계를 적용하여 O(n) 제어 메시지 오버헤드로 오더러와 모든 참여 노드들 간의 지연 값을 추정하고, 계산된 노드의 좌표 값을 기반으로 클러스터링 및 클러스터 리더 선정, 유효 대역폭 추정, 하이브리드 P2P 멀티캐스트 트리 구성 과정을 수행한 다음, 멀티 클라우드 상의 참여 노드들로 배포함으로써, 블록체인 블록 전송 시스템의 성능을 향상시켜 모든 참여 노드들에 블록을 빠르게 보낼 수 있다.

[253] 전술한 본 실시예에 따른 방법들은 다양한 컴퓨터 수단을 통해 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 컴퓨터 판독 가능 매체에 기록되는 프로그램 명령은 본 발명을 위해 특별히 설계되고 구성된 것들이거나 컴퓨터

소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다.

[254] 컴퓨터 판독 가능 매체의 예에는 롬(rom), 램(ram), 플래시 메모리(flash memory) 등과 같이 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러(compiler)에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터(interpreter) 등을 사용해서 컴퓨터에 의해 실행될 수 있는 고급 언어 코드를 포함한다. 상술한 하드웨어 장치는 본 발명의 동작을 수행하기 위해 적어도 하나의 소프트웨어 모듈로 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[255] 이상 실시예를 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

청구범위

[청구항 1] 멀티 클라우드에서의 블록 데이터 전송을 위한 유효 대역폭 추정 방법으로서,
가우시안 모델을 따르는 블록의 생성 주기에서, 다음 블록의 생성 전에 기설정된 성공 확률로 이전 블록의 전송을 완료하기 위한 블록 전송 시간을 계산하는 단계; 및
상기 블록에 포함되는 트랜잭션 개수와 트랜잭션 크기와 블록 헤더 크기 및 상기 블록 전송 시간에 기초하여 유효 대역폭을 추정하는 단계;
를 포함하는 블록 데이터 전송을 위한 유효 대역폭 추정 방법.

[청구항 2] 청구항 1에 있어서,
상기 가우시안 모델을 따르는 블록의 생성 주기에서 상기 블록 데이터 전송을 위한 트랜잭션들이 기하급수적으로 분포하고, 상기 멀티 클라우드의 오더러(orderer)에서 블록이 고정된 크기로 생성되는, 블록 데이터 전송을 위한 유효 대역폭 추정 방법.

[청구항 3] 청구항 2에 있어서,
상기 블록은 미리 설정된 개수의 트랜잭션들이 오더러에 도착할 때 생성되는, 블록 데이터 전송을 위한 유효 대역폭 추정 방법.

[청구항 4] 청구항 3에 있어서,
상기 블록에 포함된 인접한 두 트랜잭션들 사이의 시간 간격은 기하급수적으로 분포하는 것으로 설정되는, 블록 데이터 전송을 위한 유효 대역폭 추정 방법.

[청구항 5] 청구항 4에 있어서,
상기 블록에 포함된 트랜잭션 개수가 미리 설정된 기준값 이상으로 클 때, 상기 트랜잭션들이 도착할 때까지의 시간 간격은 상기 가우시안 모델로 정의되는, 블록 데이터 전송을 위한 유효 대역폭 추정 방법.

[청구항 6] 청구항 5에 있어서,
상기 가우시안 모델(G)은 수학식 12로 정의되고,
[수학식 12]

$$G(N_{trx}^{blk}/\lambda, \sqrt{N_{trx}^{blk}}/\lambda)$$

여기서, N_{trx}^{blk}는 상기 트랜잭션 개수를 나타내고, λ는 지수 분포 확률 모델(exponentially distributed stochastic model)의 트랜잭션 속도인, 블록 데이터 전송을 위한 유효 대역폭 추정 방법.

[청구항 7] 청구항 6에 있어서,
상기 블록 전송 시간은 수학식 13으로 정의되고,
[수학식 13]

$$(N_{trx}^{blk} - k \cdot \sqrt{N_{trx}^{blk}})/\lambda$$

여기서, k는 가중치로서 상기 성공 확률에 대응하여 고정되는, 블록 데이터 전송을 위한 유효 대역폭 추정 방법.

[청구항 8] 청구항 7에 있어서,
상기 성공 확률은 상기 블록 전송 시간이 더 작게 설정됨에 따라서 업로드 대역폭 사용을 희생하면서 더 작은 블록 버퍼링 시간을 확률적으로 보장하는, 블록 데이터 전송을 위한 유효 대역폭 추정 방법.

[청구항 9] 청구항 7에 있어서,
상기 유효 대역폭은 수학식 14로 추정되고,
[수학식 14]

$$BW_{eff} = \frac{N_{trx}^{blk} \cdot B_{TX} + B_{blk\_hd}}{\tilde{t}_{BLK}}$$

여기서, B_{TX}는 상기 트랜잭션 크기이고, B_{blk_hd}는 상기 블록 헤더 크기이며, $\tilde{t}_{BLK}$는 상기 트랜잭션들이 도착할 때까지의 시간 간격인, 블록 데이터 전송을 위한 유효 대역폭 추정 방법.

[청구항 10] 청구항 9에 있어서,
상기 트랜잭션 개수가 소정의 기준값보다 클 때, 상기 유효 대역폭은 상기 지수 분포 확률 모델의 트랜잭션 속도에 따라 선형적으로 증가하는, 블록 데이터 전송을 위한 유효 대역폭 추정 방법.

[청구항 11] 청구항 10에 있어서,
상기 지수 분포 확률 모델의 트랜잭션 속도가 소정의 기준값보다 크면, 상기 유효 대역폭은 상기 트랜잭션 개수에 대해 기하급수적으로 감소하는, 블록 데이터 전송을 위한 유효 대역폭 추정 방법.

[청구항 12] 청구항 11에 있어서,
상기 성공 확률이 커질수록 상기 트랜잭션 개수에 대한 기하급수적 감소율은 더 커지는, 블록 데이터 전송을 위한 유효 대역폭 추정 방법.

[청구항 13] 청구항 9에 있어서,
상기 트랜잭션 개수가 미리 결정된 트랜잭션 개수보다 작더라도, 상기 트랜잭션들 중 가장 오래된 트랜잭션의 버퍼링 시간이 소정의 임계값보다 크면 그 즉시 블록을 생성하여, 상기 트랜잭션들이 도착할 때까지의 시간 간격을 미리 설정된 크기보다 작게 유지하는 단계를 더 포함하는, 블록 데이터 전송을 위한 유효 대역폭 추정 방법.

[청구항 14] 멀티 클라우드에서의 블록 데이터 전송을 위한 유효 대역폭 추정 장치로서,
프로세서; 및
상기 프로세서에 연결되고 상기 프로세서에 의해 실행되는 적어도 하나의 명령을 저장하는 메모리를 포함하고,
상기 적어도 하나의 명령은, 상기 프로세서에 의해,

가우시안 모델을 따르는 블록의 생성 주기에서, 다음 블록의 생성 전에 기설정된 성공 확률로 이전 블록의 전송을 완료하기 위한 블록 전송 시간을 계산하는 단계, 및

상기 블록에 포함되는 트랜잭션 개수와 트랜잭션 크기와 블록 헤더 크기 및상기 블록 전송 시간에 기초하여 유효 대역폭을 추정하는 단계를 수행하도록 하는, 블록 데이터 전송을 위한 유효 대역폭 추정 장치.

[청구항 15] 청구항 14에 있어서,

상기 적어도 하나의 명령은, 상기 프로세서에 의해,

상기 계산하는 단계에서 상기 가우시안 모델을 따르는 블록의 생성 주기에서 상기 블록 데이터 전송을 위한 트랜잭션들은 기하급수적으로 분포하고 상기 멀티 클라우드의 오더러(orderer)에서 블록이 고정된 크기로 생성되도록 처리하는, 블록 데이터 전송을 위한 유효 대역폭 추정 장치.

[청구항 16] 청구항 15에 있어서,

상기 적어도 하나의 명령은, 상기 프로세서에 의해,

상기 계산하는 단계에서 미리 설정된 개수의 트랜잭션들이 오더러에 도착할 때 상기 블록을 생성하는 단계를 더 수행하도록 하는, 블록 데이터 전송을 위한 유효 대역폭 추정 장치.

[청구항 17] 청구항 16에 있어서,

상기 적어도 하나의 명령은, 상기 프로세서에 의해,

상기 블록에 포함된 인접한 두 트랜잭션들 사이의 시간 간격을 기하급수적으로 분포하는 것으로 처리하도록 하는, 블록 데이터 전송을 위한 유효 대역폭 추정 장치.

[청구항 18] 청구항 14에 있어서,

상기 블록에 포함된 트랜잭션 개수가 미리 설정된 기준값 이상으로 클 때, 트랜잭션들이 도착할 때까지의 시간 간격은 상기 가우시안 모델로 정의되는, 블록 데이터 전송을 위한 유효 대역폭 추정 장치.

[청구항 19] 청구항 14에 있어서,

상기 성공 확률은 상기 블록 전송 시간이 더 작게 설정됨에 따라서 업로드 대역폭 사용을 희생하면서 더 작은 블록 버퍼링 시간을 확률적으로 보장하도록 설정되는, 블록 데이터 전송을 위한 유효 대역폭 추정 장치.

[청구항 20] 청구항 14에 있어서,

상기 적어도 하나의 명령은, 상기 프로세서에 의해,

상기 블록에 포함된 트랜잭션들의 개수가 미리 결정된 트랜잭션 개수보다 작더라도, 상기 트랜잭션들 중 가장 오래된 트랜잭션의 버퍼링 시간이 소정의 임계값보다 크면 그 즉시 블록을 생성하는 단계를 더 수행하도록 하여, 상기 트랜잭션들이 도착할 때까지의 시간 간격을 미리 설정된 크기보다 작게 유지하는, 블록 데이터 전송을 위한 유효 대역폭

추정 장치.

[청구항 21] 멀티 클라우드의 확장 가능한 하이퍼레저 패브릭 블록체인을 위한 블록 전송 시스템의 제어 메시지 생성 방법으로서,
복수의 참여 노드들을 복수의 클러스터들에 속한 노드들로 구분하는 단계;
상기 복수의 참여 노드들 중에서 상기 복수의 클러스터들 각각의 클러스터 리더를 선택하는 단계; 및
상기 복수의 클러스터들과 상기 클러스터 리더에 대한 정보에 기초하여 하이브리드 P2P(peer to peer) 멀티캐스트 트리(multicast tree)를 구성하는 단계를 포함하고,
상기 구성하는 단계는, 상기 하이브리드 P2P 멀티캐스트 트리에 포함되는 각 참여 노드의 트리 정보를 노드 IP(internet protocol) 필드, 자식 노드 수(the number of child nodes) 필드 및 자식 노드 목록의 오프셋(offset of child nodes lists) 필드의 조합으로 형성하는 단계를 포함하는, 제어 메시지 생성 방법.

[청구항 22] 청구항 21에 있어서,
상기 노드 IP 필드는 현재 노드의 IP 주소를 나타내고, 상기 자식 노드 수 필드는 상기 하이브리드 P2P 멀티캐스트 트리를 통해 상기 현재 노드에 연결된 자식 노드의 수를 나타내고, 상기 자식 노드 목록의 오프셋 필드는 제어 메시지 자체 내에서 첫 번째 자식 노드의 위치를 나타내는, 제어 메시지 생성 방법.

[청구항 23] 청구항 21에 있어서,
상기 구성하는 단계는, 상기 각 참여 노드의 트리 정보를 미리 설정된 순서대로 일렬로 배열한 페이로드를 형성하는 단계를 더 포함하는 제어 메시지 생성 방법.

[청구항 24] 청구항 23에 있어서,
상기 복수의 참여 노드들 중 상기 페이로드를 포함한 제어 메시지를 받은 특정 참여 노드가 하위 트리에 대한 정보를 제거하여 상기 제어 메시지를 재구성하는 단계를 더 포함하는 제어 메시지 생성 방법.

[청구항 25] 청구항 21에 있어서,
상기 복수의 참여 노드들의 각 노드와 상기 멀티 클라우드의 오더러(orderer) 간에 추정된 엔드투엔드(end-to-end) 지연(delay)에 기초하여 획득되는 상기 복수의 참여 노드들의 유효 대역폭과 지원가능한 자식 노드들의 개수 중 적어도 어느 하나가 미리 설정된 기준 범위를 벗어난 변동이 있는지를 판단하는 단계를 더 포함하며,
상기 구성하는 단계는 상기 판단하는 단계의 판단 결과가 변동 있음인 경우에 수행되는, 제어 메시지 생성 방법.

[청구항 26] 청구항 21에 있어서,

상기 구분하는 단계, 상기 선택하는 단계 및 상기 구성하는 단계를 반복적으로 수행하여 클러스터 개수와 클러스터 리더를 최종 결정하는 단계를 더 포함하는, 제어 메시지 생성 방법.

[청구항 27] 청구항 21 내지 청구항 26 중 어느 하나의 제어 메시지 생성 방법에 이용되는 제어 메시지 포맷으로서,
헤더 및 페이로드를 포함하고,
상기 헤더는 버전(version), 메시지 유형(message type) 및 메시지 길이(message length)에 대한 정보를 담고,
상기 페이로드는 소정의 순번들에 따라 복수의 노드들의 트리 정보를 각각 순차적으로 나열한 형태를 구비하는, 제어 메시지 포맷.

[청구항 28] 청구항 27에 있어서,
상기 복수의 노드들 각각의 트리 정보는 노드 IP(internet protocol), 자식 노드 수(the number of child nodes) 및 자식 노드 목록의 오프셋(offset of child nodes lists)에 대한 필드들을 구비하는, 제어 메시지 포맷.

[청구항 29] 멀티 클라우드의 확장가능한 하이퍼레저 패브릭 블록체인을 위한 블록 전송 시스템의 트리 정보 전달 방법으로서,
멀티 클라우드에 연결된 검증 노드에서 모든 참여 노드들로부터 좌표 값들을 수신하는 단계;
오더러(orderer)로부터 트랜잭션 속도(transaction rate)를 수신하는 단계;
상기 모든 참여 노드들 중 상기 오더러에 연결되는 앵커 노드들의 업링크 대역폭을 추정하는 단계;
상기 모든 참여 노드들의 유효 대역폭과 지원가능한 자식 노드들의 개수를 계산하는 단계;
상기 모든 참여 노드들의 각 참여 노드와 상기 오더러 간의 엔드투엔드 지연을 추정하는 단계;
상기 엔드투엔드 지연에 기초하여 상기 유효 대역폭과 상기 지원가능한 자식 노드들의 개수가 미리 설정된 기준 범위를 벗어하는 변동이 있는지를 판단하는 단계;
상기 변동이 있는 경우, 상기 좌표 값들과 상기 트랜잭션 속도에 대한 정보에 기초하여 상기 모든 참여 노드들을 복수의 클러스터들로 클러스터링하는 단계;
상기 복수의 클러스터들 각각의 클러스터 리더를 선택하는 단계;
상기 복수의 클러스터와 상기 클러스터 리더를 토대로 중첩 멀티캐스트 트리를 구성하는 단계; 및
상기 중첩 멀티캐스트 트리를 제어 메시지에 담아 상기 오더러로 전송하는 단계;를 포함하는 트리 정보 전달 방법.

[청구항 30] 청구항 29에 있어서,
상기 제어 메시지는 상기 모든 참여 노드들에 전달되어 각 참여 노드에서

메시지 전달 테이블의 업데이트에 이용되는, 트리 정보 전달 방법.

[청구항 31] 청구항 29에 있어서,
상기 클러스터링하는 단계, 상기 클러스터 리더를 선택하는 단계 및 상기 중첩 멀티캐스트 트리를 구성하는 단계를 반복적으로 수행하여 클러스터 개수와 클러스터 리더를 최종적으로 결정하는 단계를 더 포함하는, 트리 정보 전달 방법.

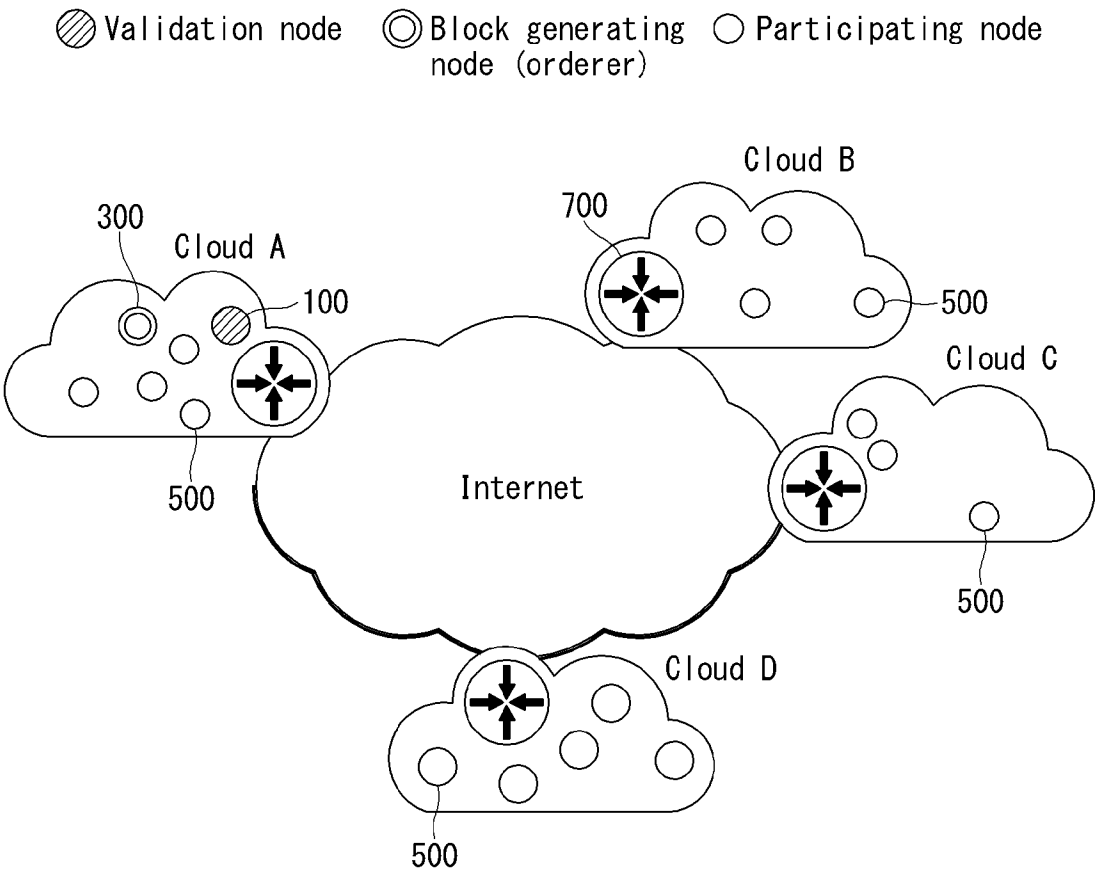
[청구항 32] 청구항 29에 있어서,
상기 제어 메시지의 포맷은 헤더 및 페이로드를 포함하고,
상기 헤더는 버전(version), 메시지 유형(message type) 및 메시지 길이(message length)에 대한 정보를 담고, 상기 페이로드는 소정의 순번들에 따라 복수의 노드들의 트리 정보를 각각 순차적으로 나열한 형태를 구비하는, 트리 정보 전달 방법.

[청구항 33] 청구항 22에 있어서,
상기 복수의 노드들의 각 노드의 트리 정보는 노드 IP(internet protocol), 자식 노드 수(the number of child nodes) 및 자식 노드 목록의 오프셋(offset of child nodes lists)에 대한 필드들을 구비하는, 트리 정보 전달 방법.

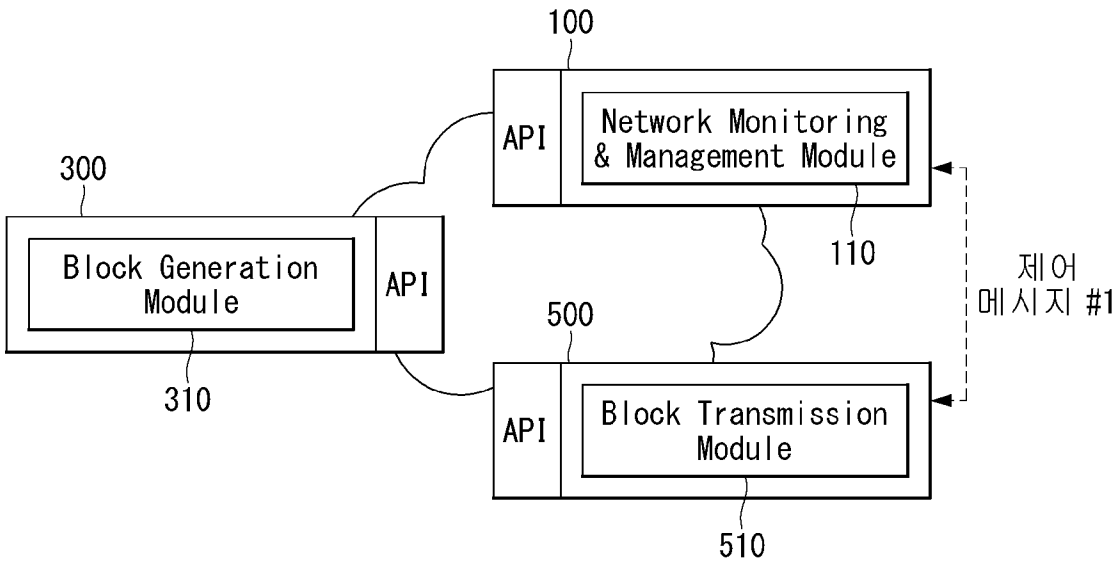
[청구항 34] 청구항 22에 있어서,
상기 복수의 참여 노드들 중 상기 페이로드를 포함한 제어 메시지를 받은 특정 참여 노드가 일부 하위 트리에 대한 정보를 제거하여 상기 제어 메시지를 재구성하는 단계를 더 포함하는 트리 정보 전달 방법.

[청구항 35] 청구항 29에 있어서,
상기 제어 메시지의 크기는 상기 중첩 멀티캐스트 트리를 통해 상기 오더러로부터 리프 노드들까지 전달되는 동안 각 부모 노드에서 단계적으로 감소되는, 트리 정보 전달 방법.

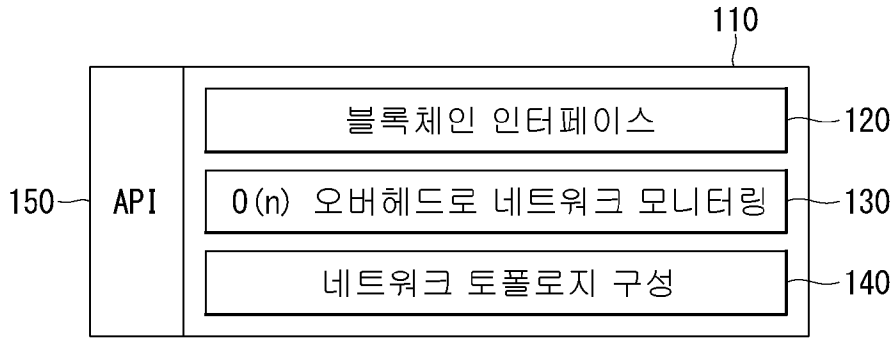
[도1]


Validation node
Block generating node (orderer)
Participating node
Cloud A
Cloud B
Cloud C
Cloud D
Internet
300
100
500
700
500
500
500

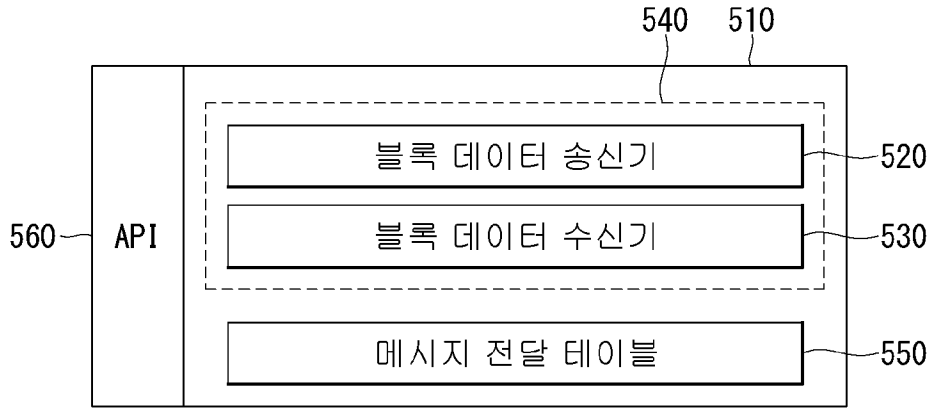
[도2]


100
API
Network Monitoring & Management Module
110
300
Block Generation Module
API
310
500
API
Block Transmission Module
510
제어 메시지 #1

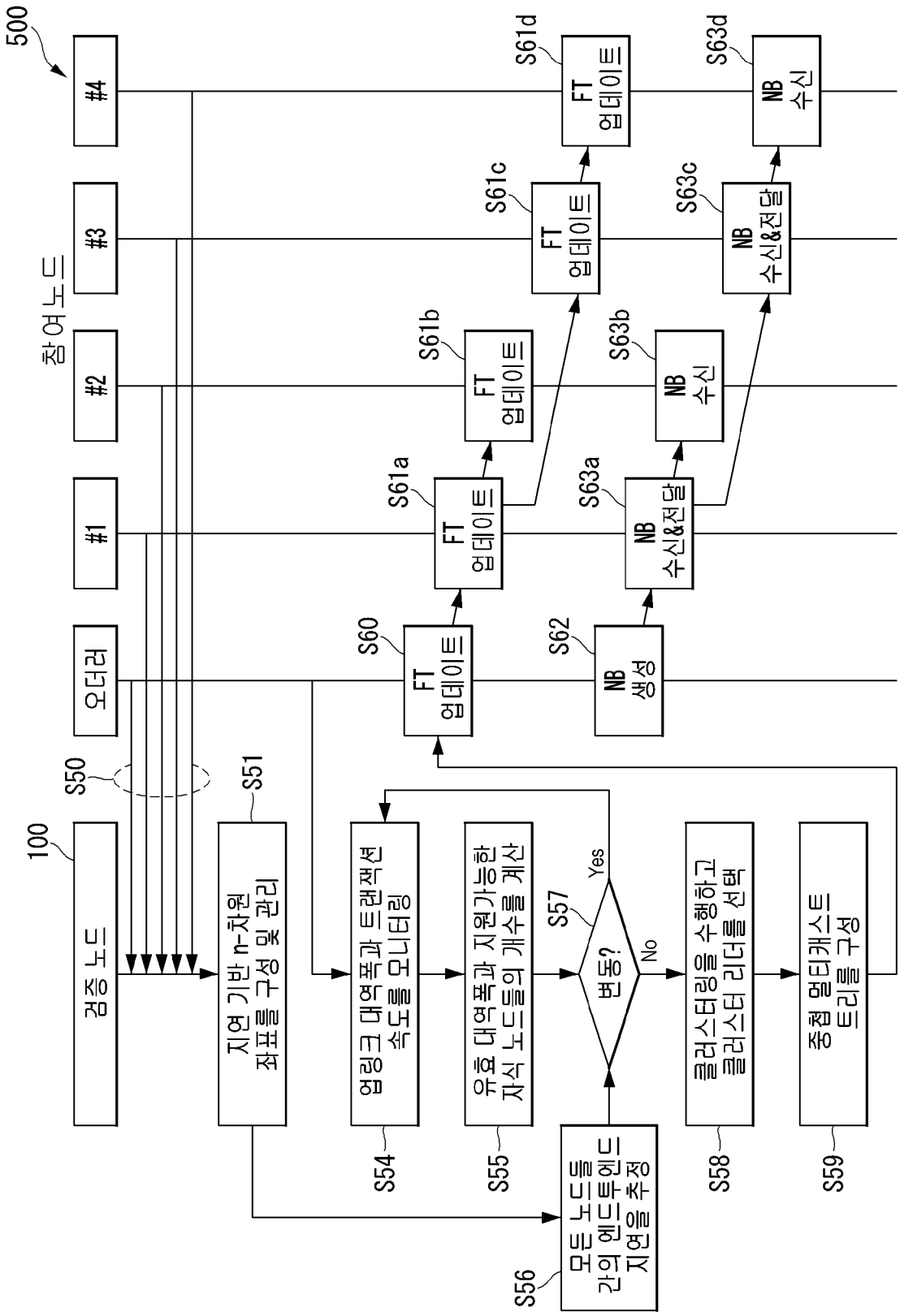
[도3]


110
블록체인 인터페이스
120
150
API
O(n) 오버헤드로 네트워크 모니터링
130
네트워크 토폴로지 구성
140

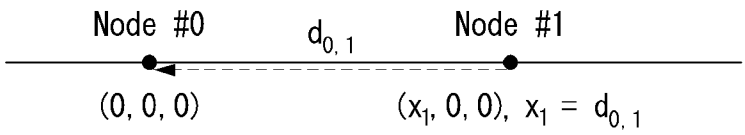
[도4]


540
510
블록 데이터 송신기
520
560
API
블록 데이터 수신기
530
메시지 전달 테이블
550

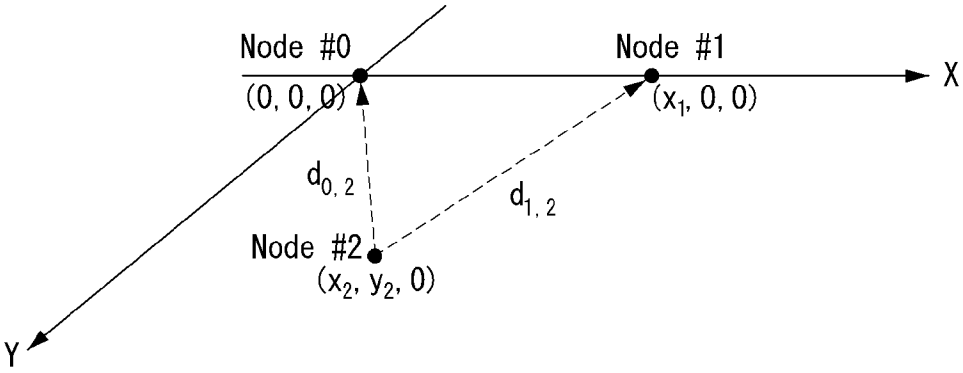
[도5]


500
참여 노드
#4
#3
#2
#1
오더러
100
검증 노드
S50
S51
지연 기반 n-차원 좌표를 구성 및 관리
S60
FT 업데이트
S61a
FT 업데이트
S61b
FT 업데이트
S61c
FT 업데이트
S61d
FT 업데이트
S62
NB 생성
S63a
NB 수신&전달
S63b
NB 수신
S63c
NB 수신&전달
S63d
NB 수신
S54
업링크 대역폭과 트랜잭션 속도를 모니터링
S55
유효 대역폭과 지원가능한 자식 노드들의 개수를 계산
S57
변동?
Yes
No
S56
모든 노드들 간의 엔드투엔드 지연을 추정
S58
클러스터링을 수행하고 클러스터 리더를 선택
S59
중첩 멀티캐스트 트리를 구성

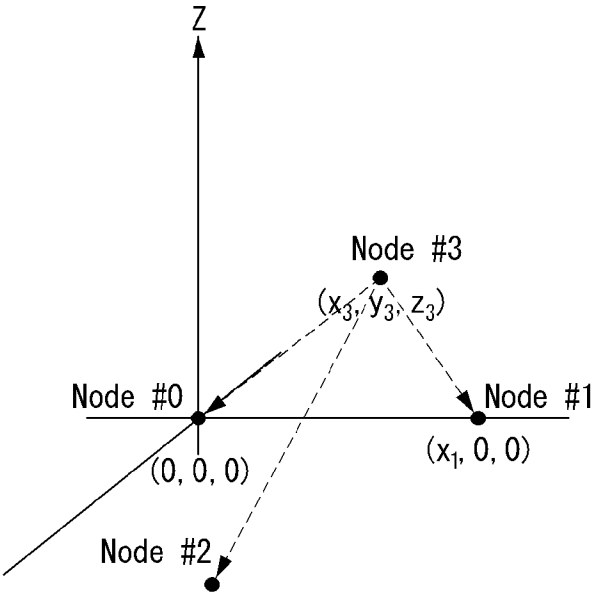
[도6]


Node #0
$d_{0,1}$
Node #1
(0, 0, 0)
(x_1, 0, 0), $x_1 = d_{0,1}$

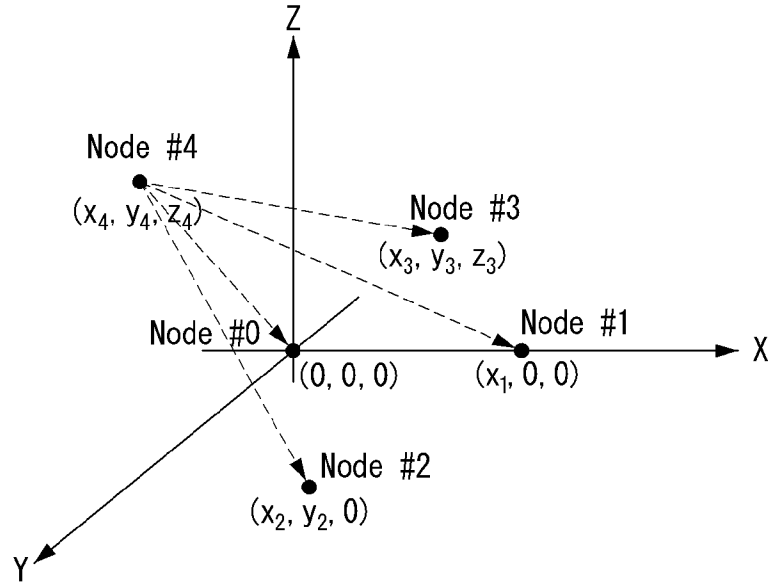
[도7]


Node #0
(0, 0, 0)
Node #1
(x1, 0, 0)
X
d0, 2
d1, 2
Node #2
(x2, y2, 0)
Y

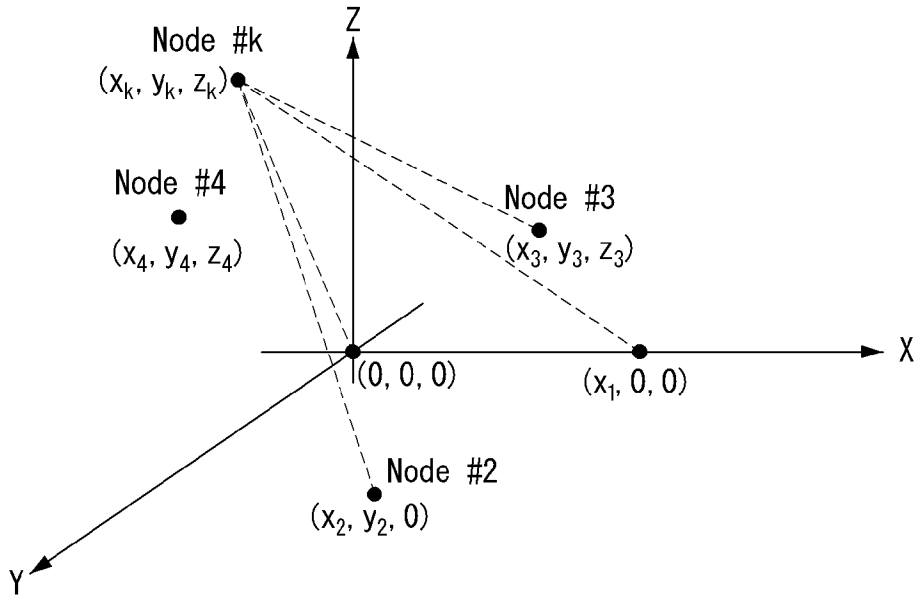
[도8]


Z
Node #3
(x3, y3, z3)
Node #0
Node #1
(0, 0, 0)
(x1, 0, 0)
Node #2

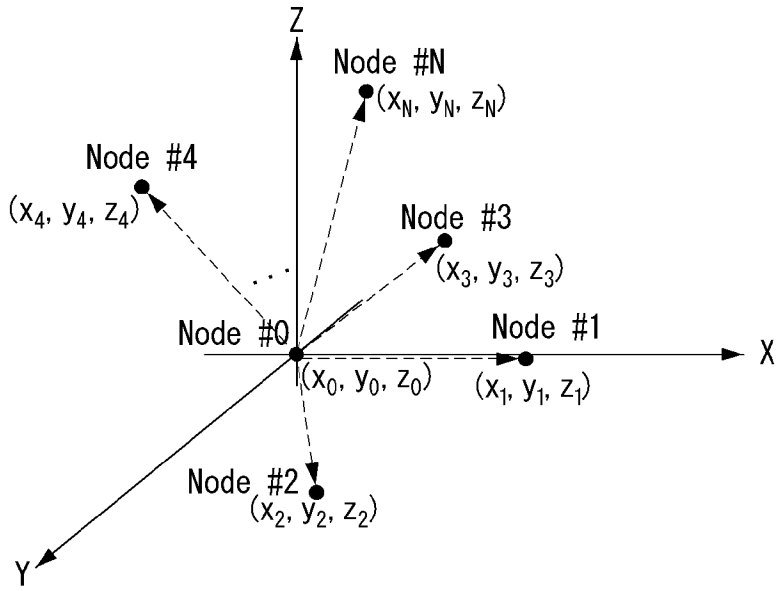
[도9]


Z
Node #4
(x4, y4, z4)
Node #3
(x3, y3, z3)
Node #0
Node #1
X
(0, 0, 0)
(x1, 0, 0)
Node #2
(x2, y2, 0)
Y

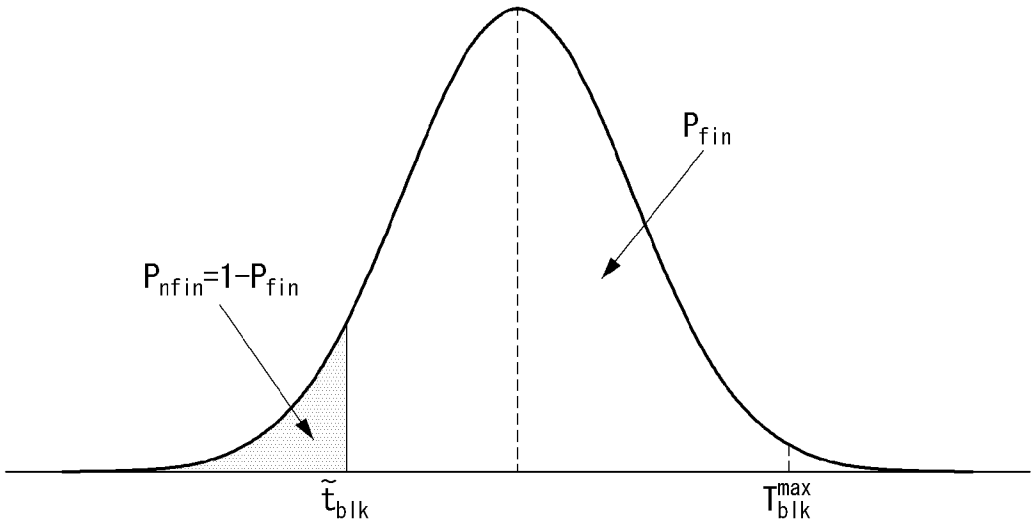
[도10]


Z
Node #k
(x_k, y_k, z_k)
Node #4
(x_4, y_4, z_4)
Node #3
(x_3, y_3, z_3)
X
(0, 0, 0)
$(x_1, 0, 0)$
Node #2
$(x_2, y_2, 0)$
Y

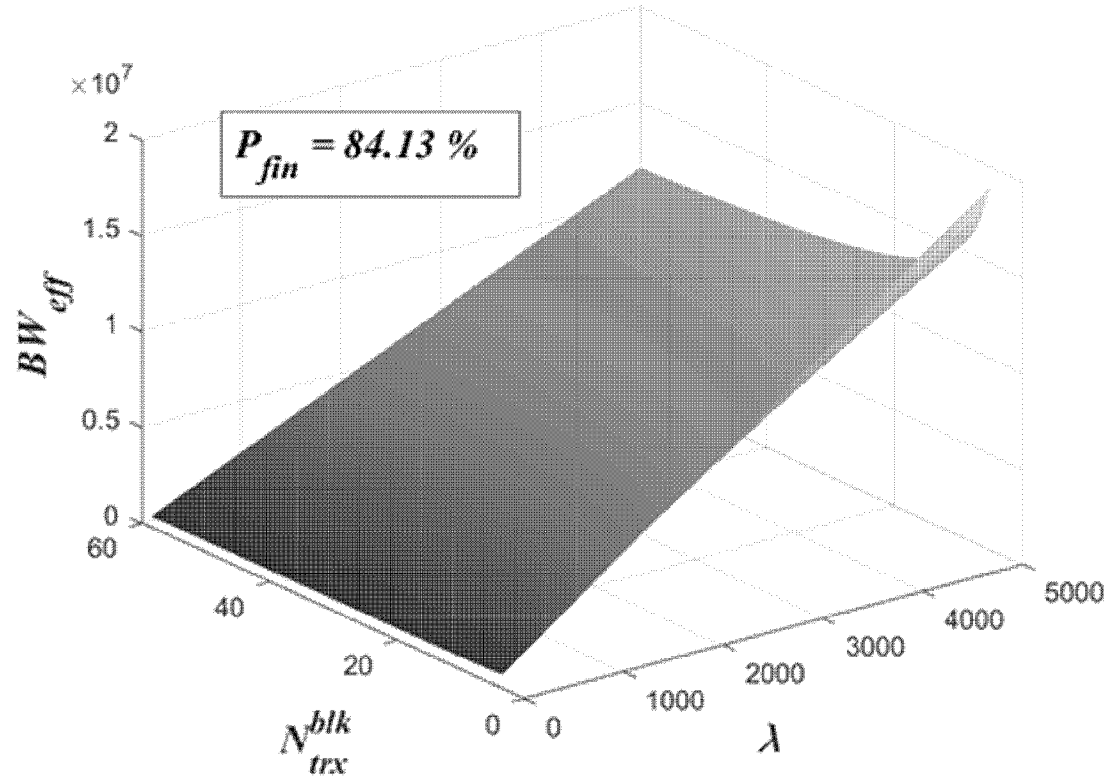
[도11]


Z
Node #N
(x_N, y_N, z_N)
Node #4
(x_4, y_4, z_4)
Node #3
(x_3, y_3, z_3)
Node #0
Node #1
X
(x_0, y_0, z_0)
(x_1, y_1, z_1)
Node #2
(x_2, y_2, z_2)
Y

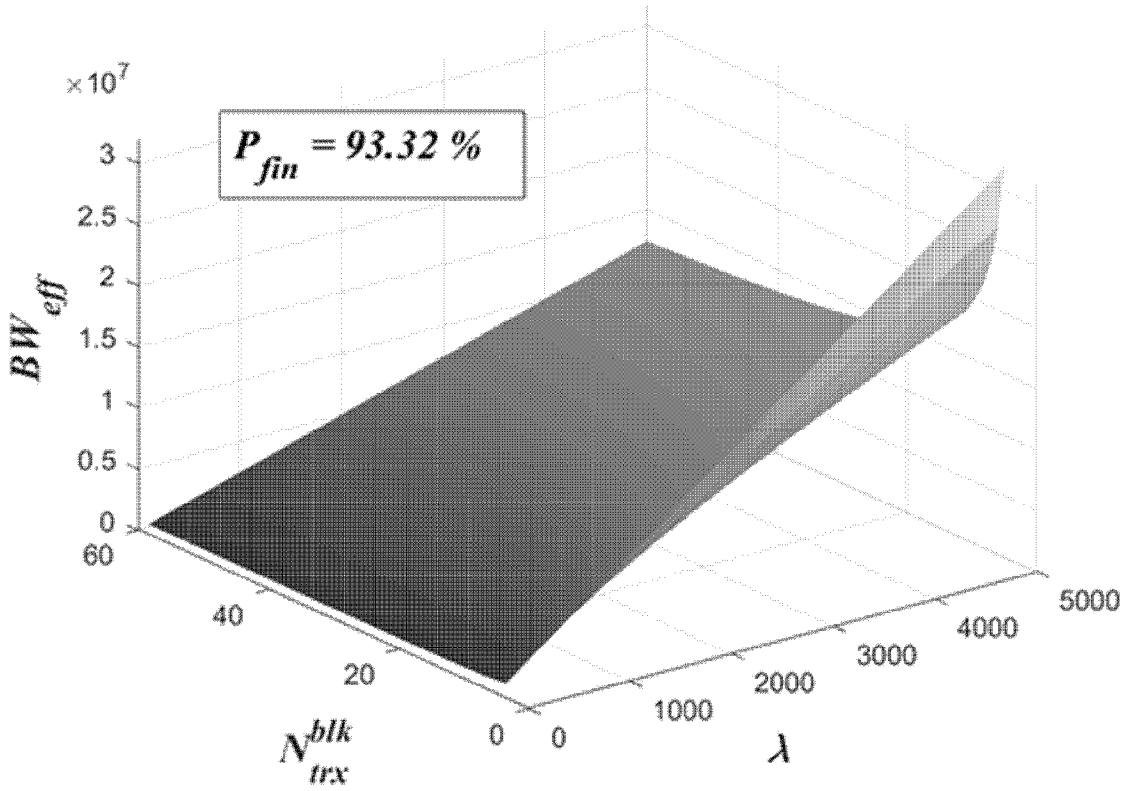
[도12]


P_{fin}
$P_{nfin}=1-P_{fin}$
$\tilde{t}_{blk}$
T_{blk}^{max}

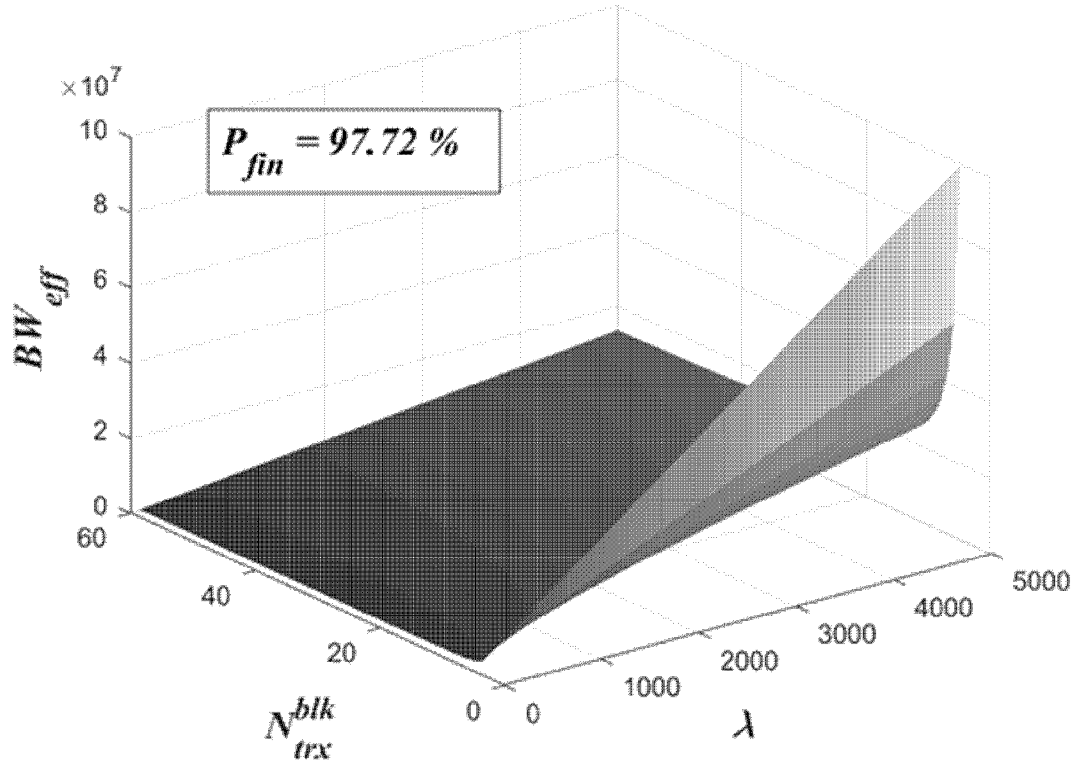
[도13]


×10^7
2
1.5
1
0.5
0
BW_eff
P_fin = 84.13 %
60
40
20
0
N_trx^blk
0
1000
2000
3000
4000
5000
λ

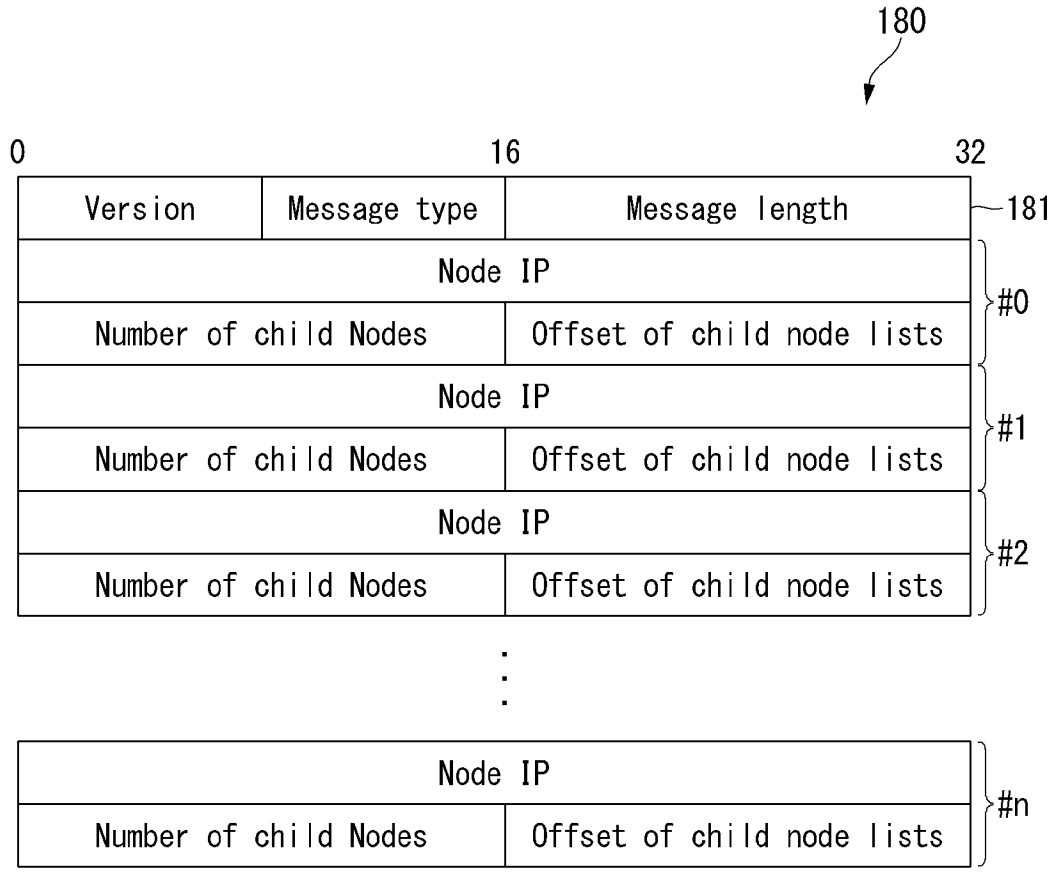
[도14]


×10^7
3
2.5
2
1.5
1
0.5
0
BW_eff
P_fin = 93.32 %
60
40
20
0
N_trx^blk
0
1000
2000
3000
4000
5000
λ

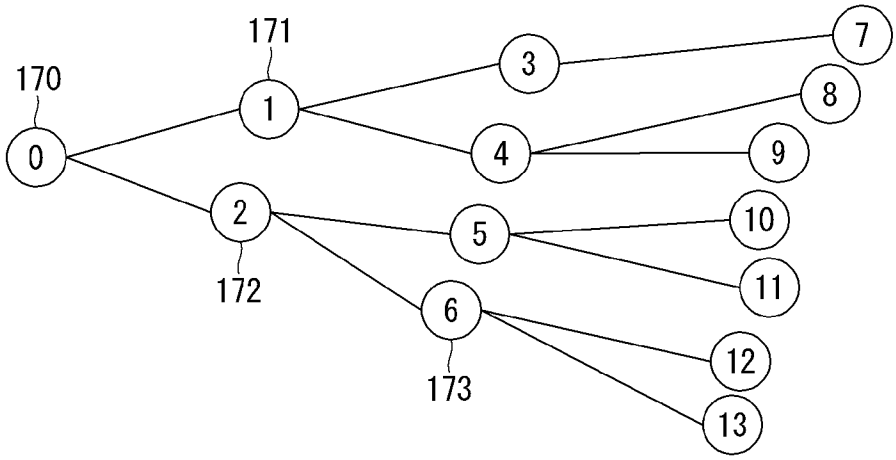
[도15]


×10^7
10
8
6
4
2
0
BW_eff
P_fin = 97.72 %
60
40
20
0
N_trx^blk
0
1000
2000
3000
4000
5000
λ

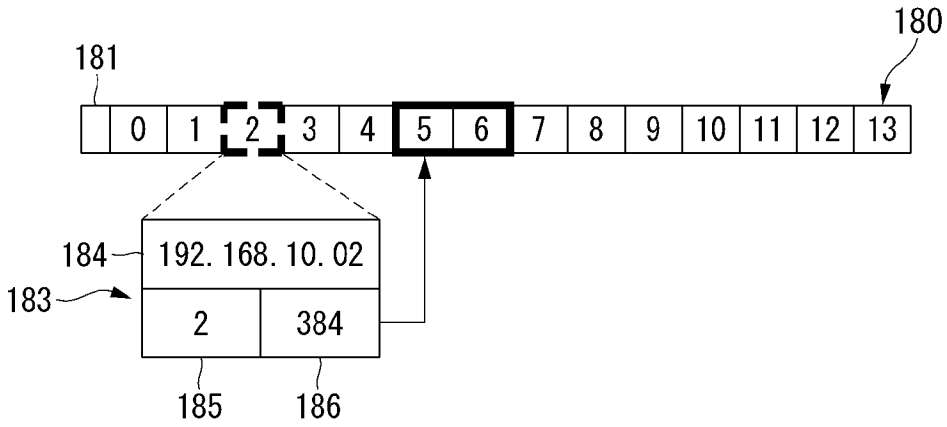
[도16]


180
0
16
32
Version
Message type
Message length
181
Node IP
Number of child Nodes
Offset of child node lists
#0
Node IP
Number of child Nodes
Offset of child node lists
#1
Node IP
Number of child Nodes
Offset of child node lists
#2
Node IP
Number of child Nodes
Offset of child node lists
#n

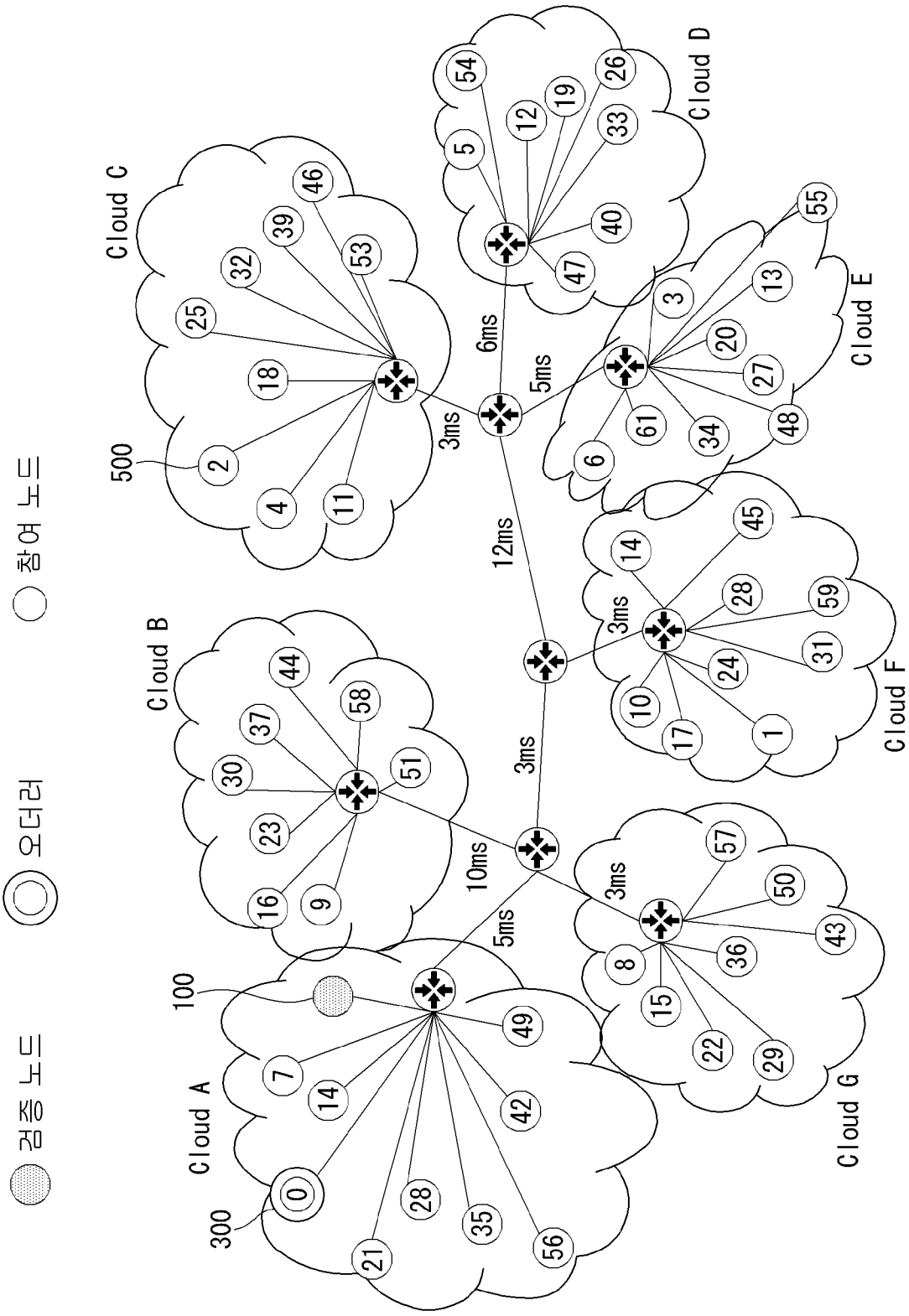
[도17]


170
171
172
173
0
1
2
3
4
5
6
7
8
9
10
11
12
13

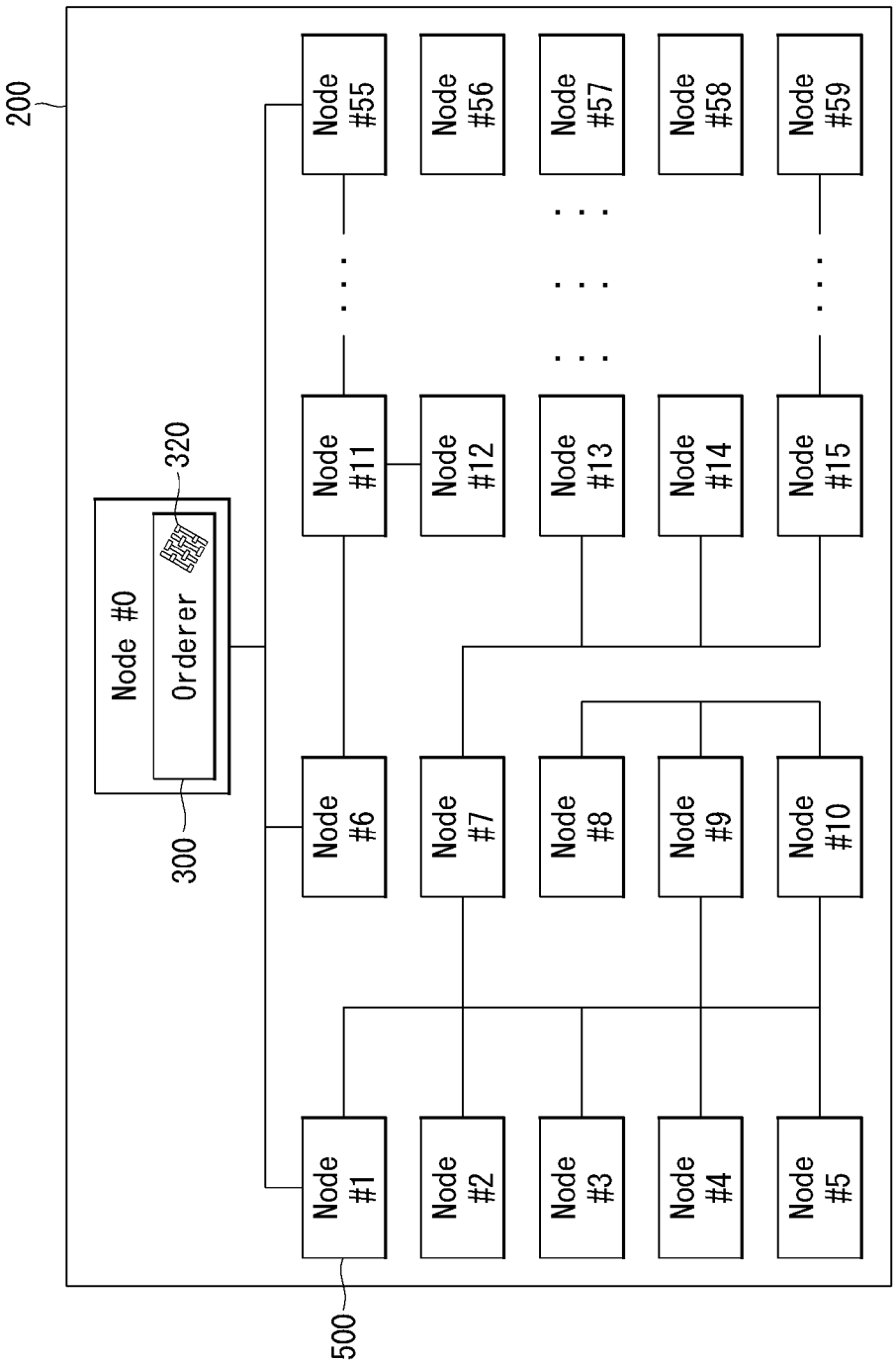
[도18]


180
181
0
1
2
3
4
5
6
7
8
9
10
11
12
13
184
192. 168. 10. 02
183
2
384
185
186

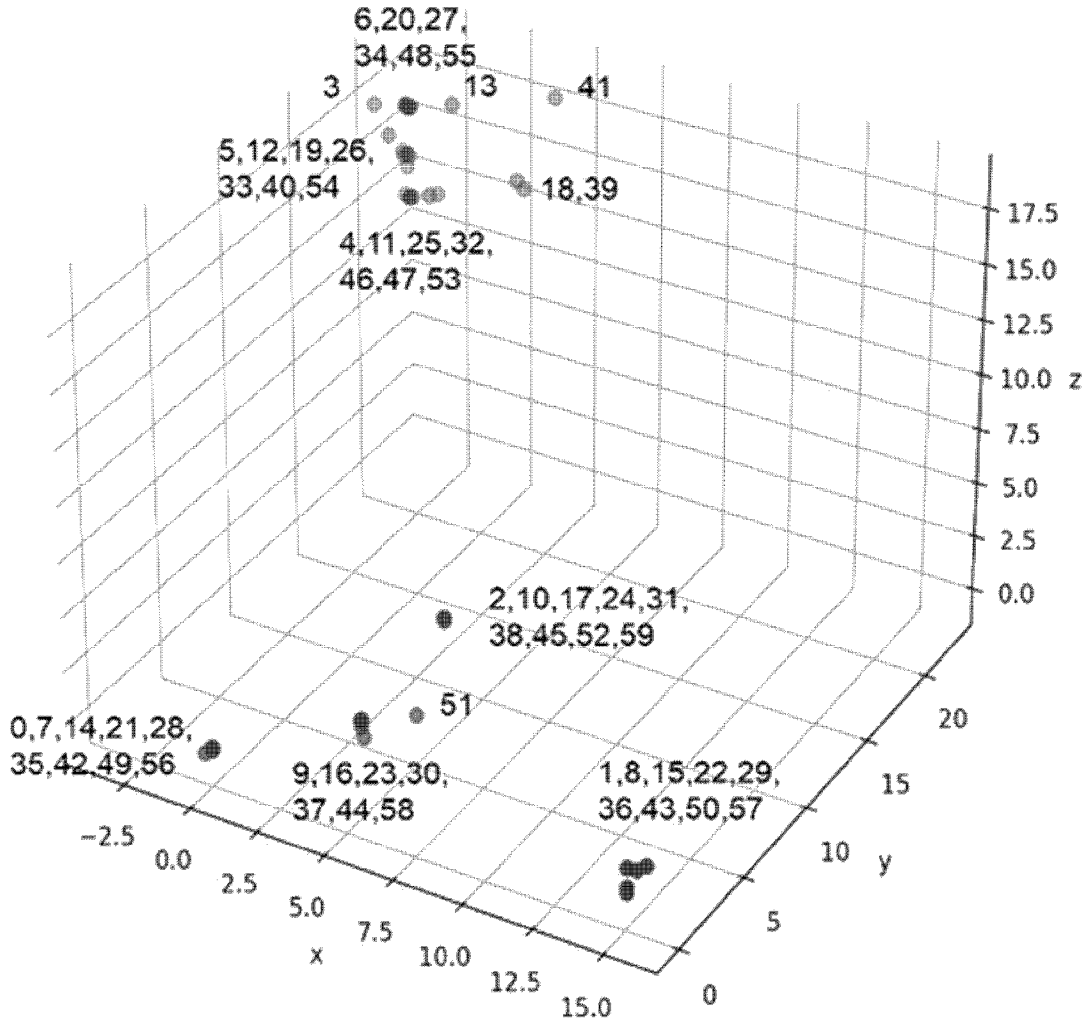
[도19]


검증 노드
오더러
참여 노드
Cloud A
Cloud B
Cloud C
Cloud D
Cloud E
Cloud F
Cloud G
100
300
500
3ms
5ms
6ms
12ms
10ms

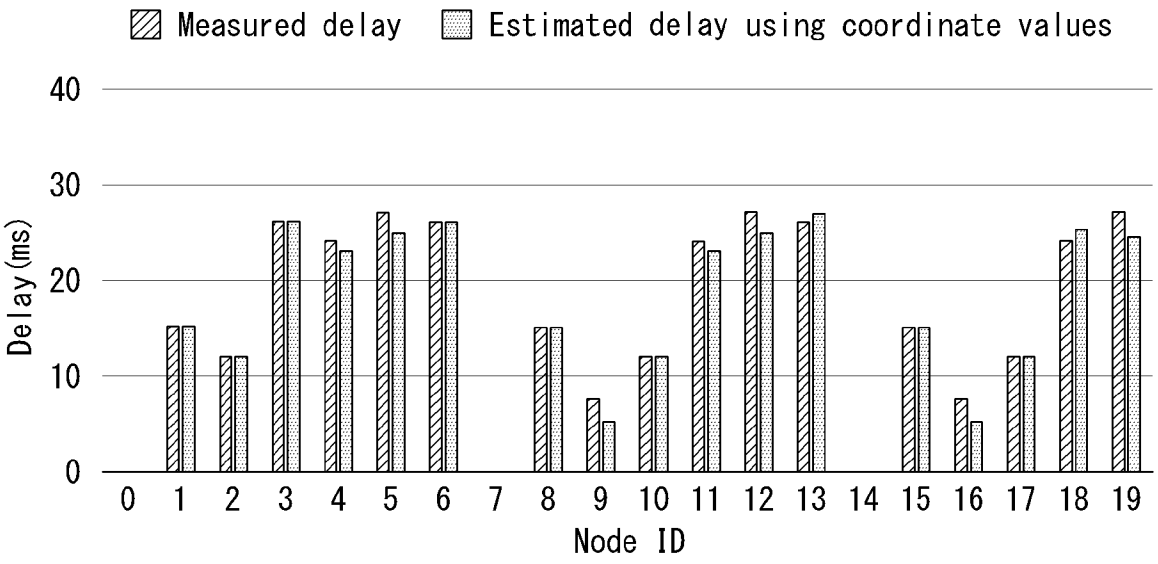
[도20]


200
Node #55
Node #56
Node #57
Node #58
Node #59
320
Node #0
Orderer
Node #11
Node #12
Node #13
Node #14
Node #15
300
Node #6
Node #7
Node #8
Node #9
Node #10
Node #1
Node #2
Node #3
Node #4
Node #5
500

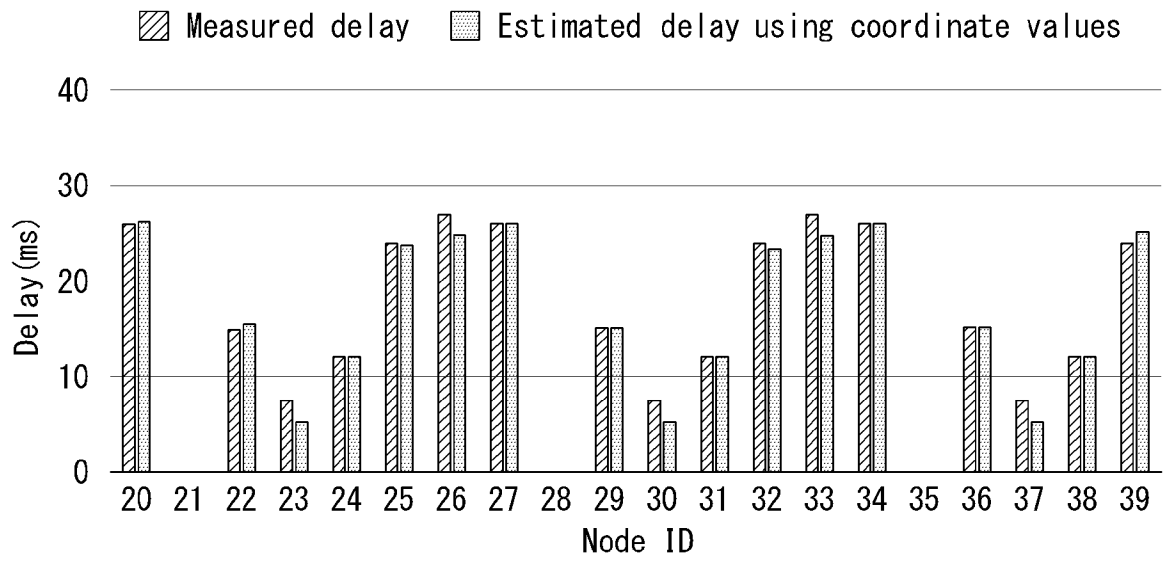
[도21]


6,20,27,
34,48,55
3
13
41
5,12,19,26,
33,40,54
18,39
4,11,25,32,
46,47,53
2,10,17,24,31,
38,45,52,59
51
0,7,14,21,28,
35,42,49,56
9,16,23,30,
37,44,58
1,8,15,22,29,
36,43,50,57
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
z
20
15
10
5
0
y
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
x

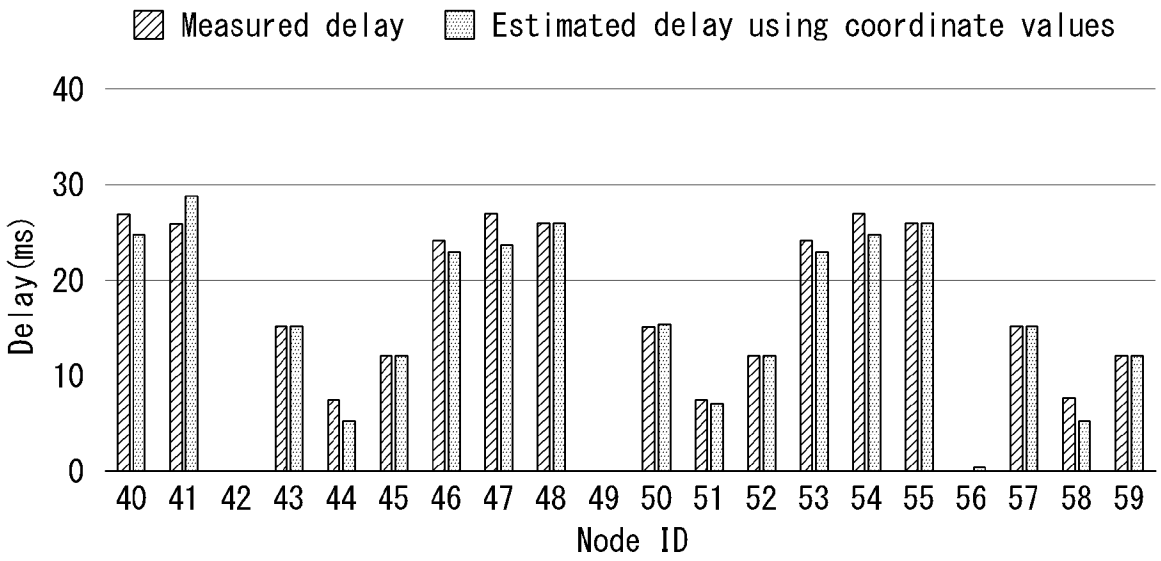
[도22a]


Measured delay
Estimated delay using coordinate values
40
30
20
10
0
Delay(ms)
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
Node ID

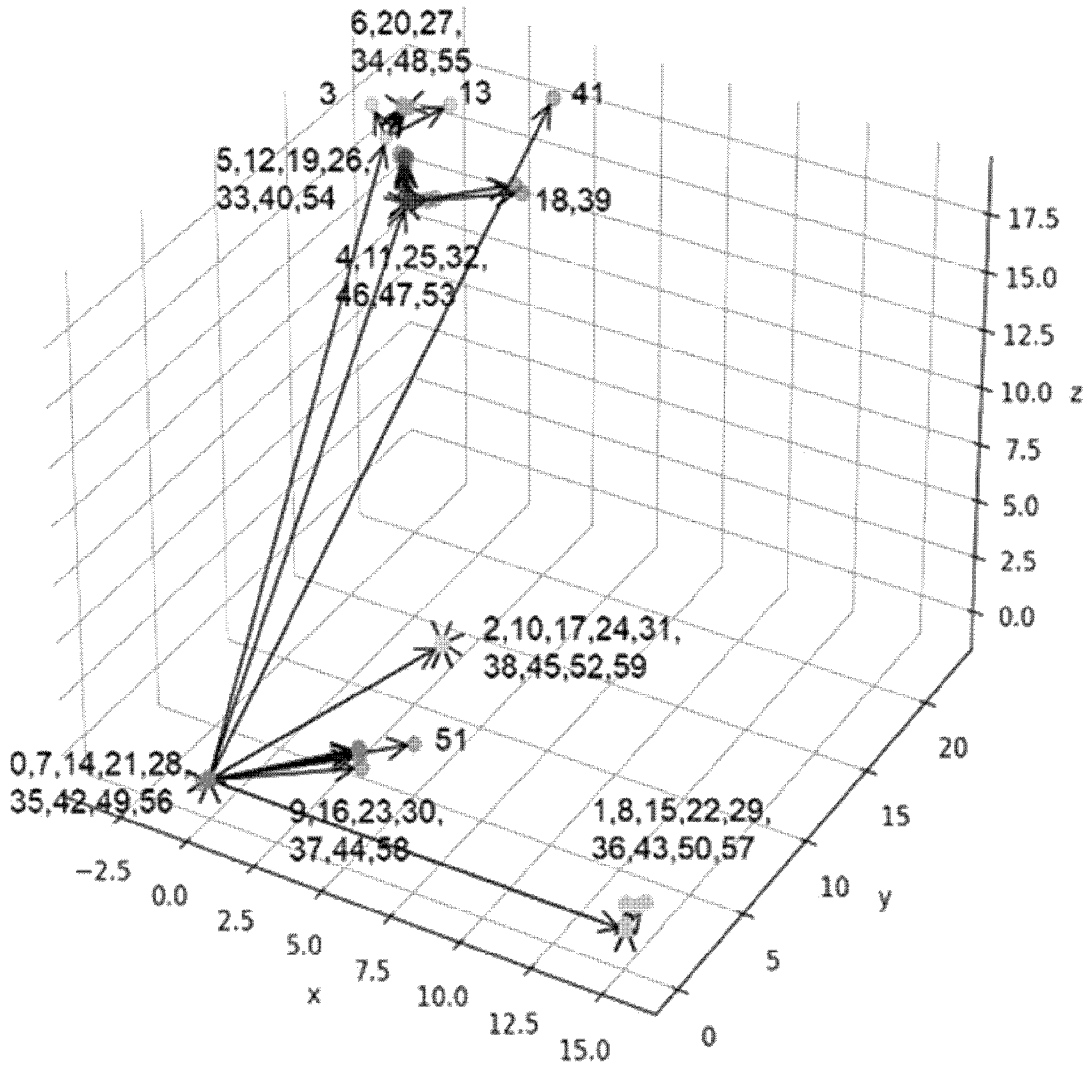
[도22b]


Measured delay
Estimated delay using coordinate values
40
30
20
10
0
Delay(ms)
20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
Node ID

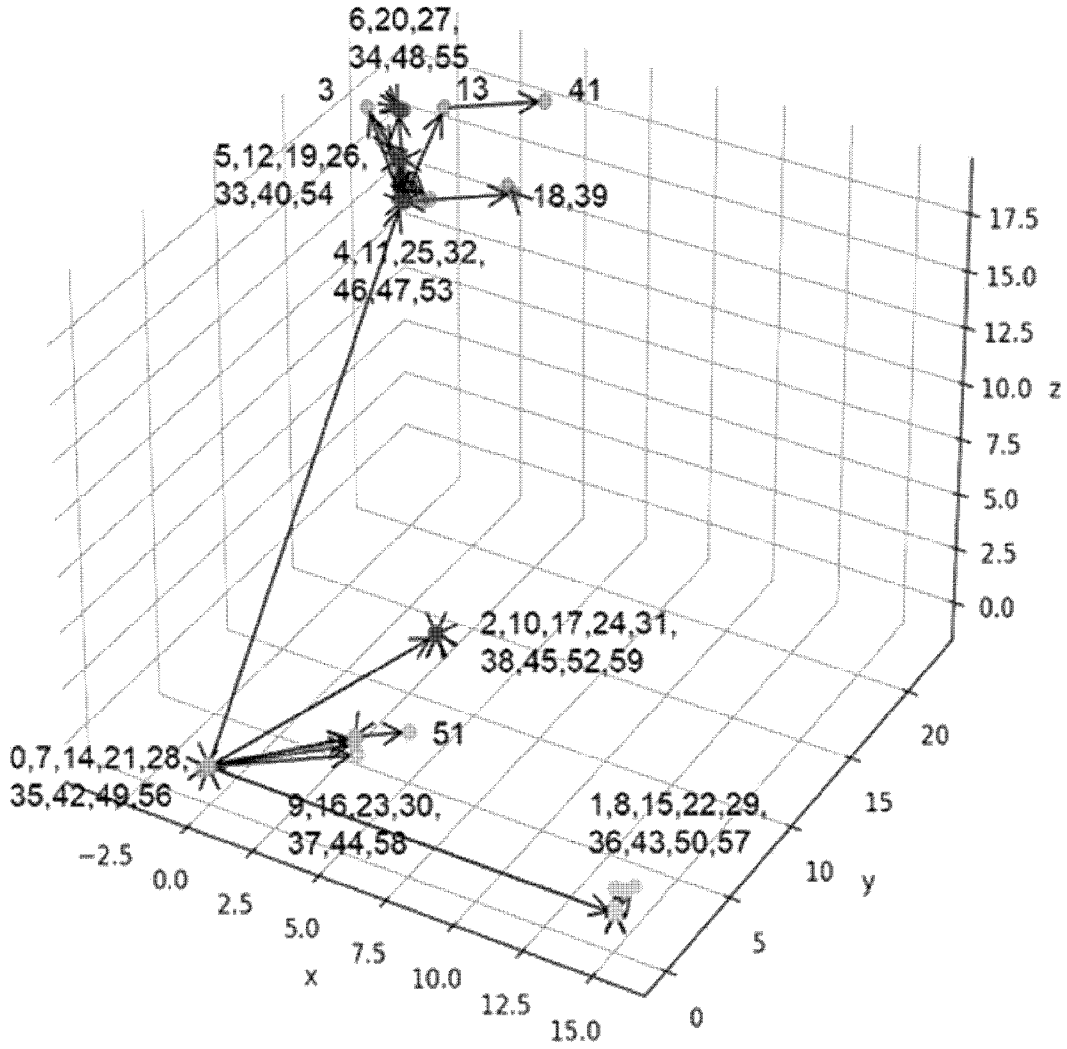
[도22c]


Measured delay
Estimated delay using coordinate values
40
30
20
10
0
Delay(ms)
40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59
Node ID

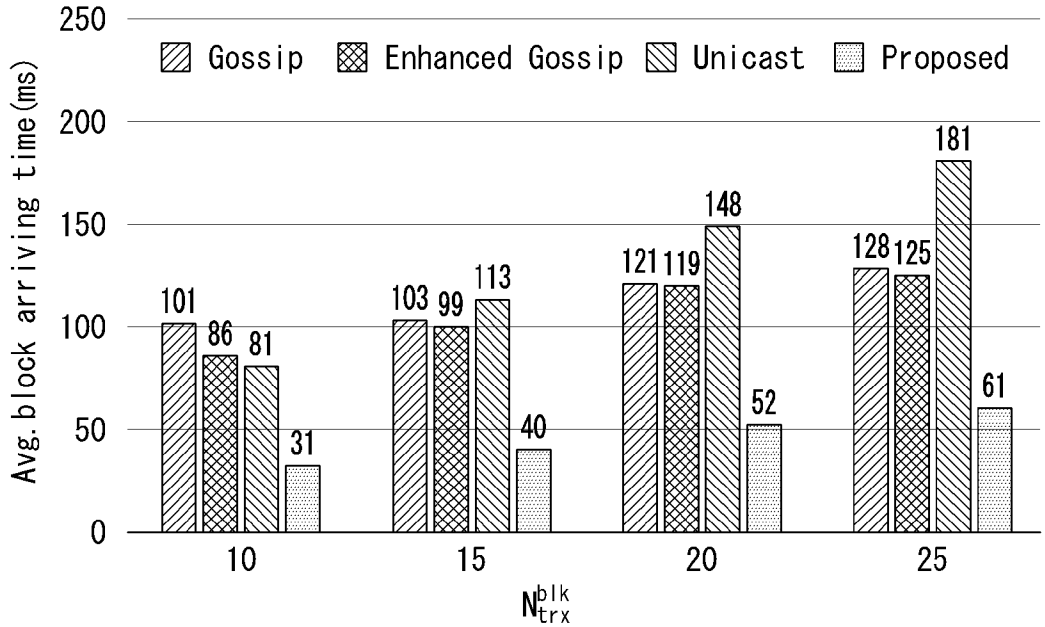
[도23]


6,20,27,
34,48,55
3
13
41
5,12,19,26,
33,40,54
18,39
4,11,25,32,
46,47,53
2,10,17,24,31,
38,45,52,59
51
0,7,14,21,28,
35,42,49,56
9,16,23,30,
37,44,58
1,8,15,22,29,
36,43,50,57
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
z
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
x
0
5
10
15
20
y

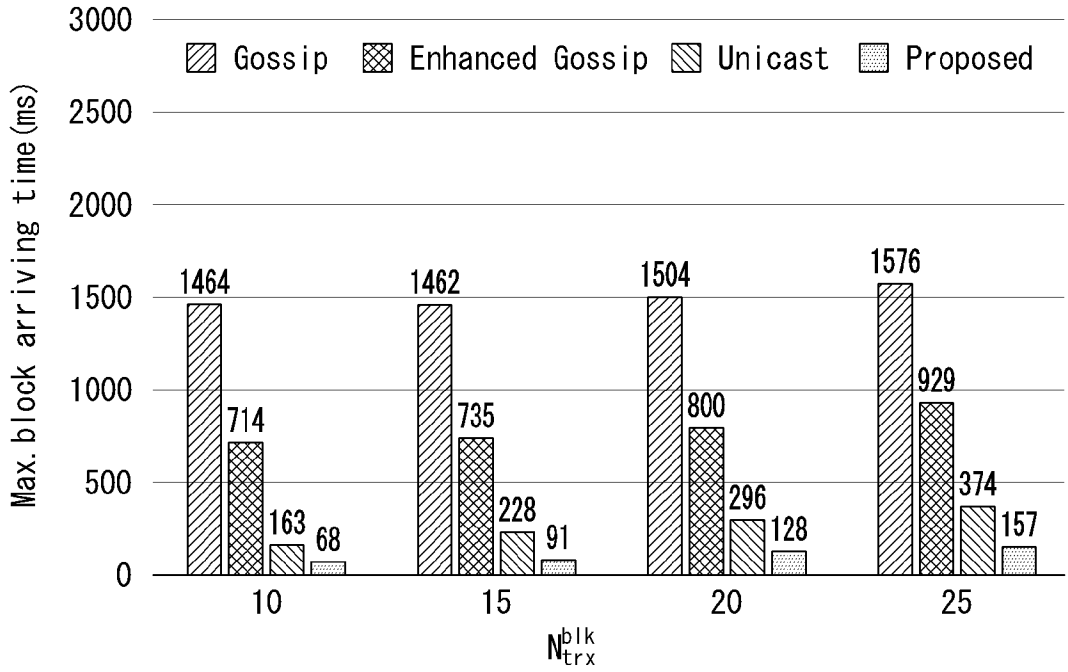
[도24]


6,20,27,
34,48,55
3
13
41
5,12,19,26,
33,40,54
18,39
4,11,25,32,
46,47,53
2,10,17,24,31,
38,45,52,59
51
0,7,14,21,28,
35,42,49,56
9,16,23,30,
37,44,58
1,8,15,22,29,
36,43,50,57
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
z
20
15
10
5
0
y
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
x

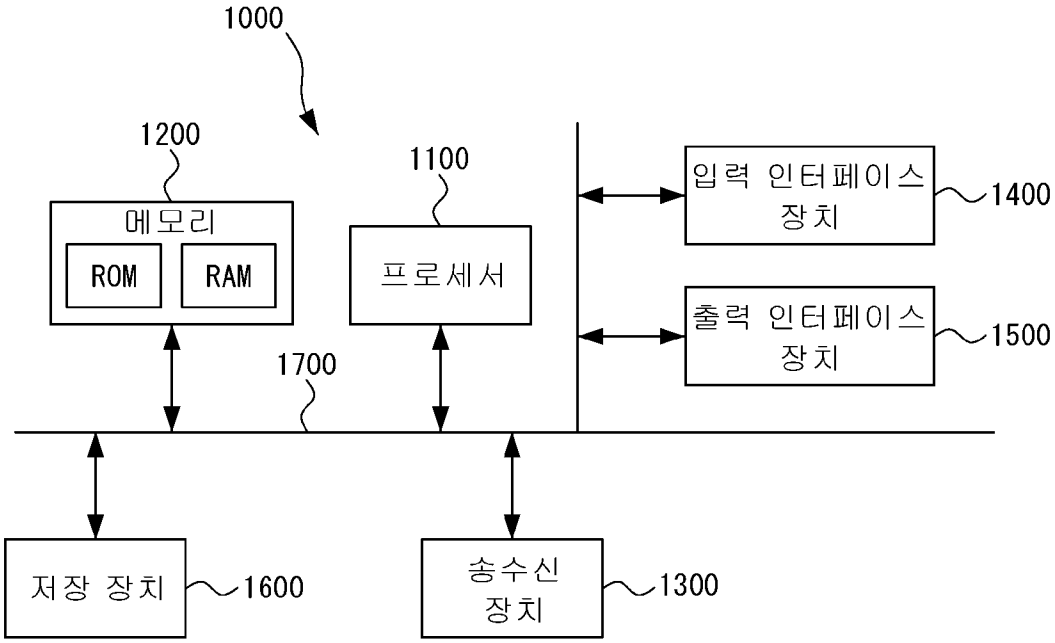
[도25]


Gossip
Enhanced Gossip
Unicast
Proposed
Avg. block arriving time(ms)
250
200
150
100
50
0
101
86
81
31
103
99
113
40
121
119
148
52
128
125
181
61
10
15
20
25
N_{trx}^{blk}

[도26]


3000
2500
2000
1500
1000
500
0
Max. block arriving time(ms)
Gossip
Enhanced Gossip
Unicast
Proposed
1464
714
163
68
1462
735
228
91
1504
800
296
128
1576
929
374
157
10
15
20
25
N_{trx}^{blk}

[도27]


1000
1200
메모리
ROM
RAM
1100
프로세서
입력 인터페이스 장치
1400
출력 인터페이스 장치
1500
1700
저장 장치
1600
송수신 장치
1300

INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR2022/015119

A. CLASSIFICATION OF SUBJECT MATTER

H04L 67/1074(2022.01)i; **H04L 43/0852**(2022.01)i; **H04L 43/0876**(2022.01)i; **H04L 67/1042**(2022.01)i; **H04L 67/1097**(2022.01)i; **H04L 12/18**(2006.01)i; **H04L 69/22**(2022.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L 67/1074(2022.01); G06F 9/50(2006.01); H04B 7/26(2006.01); H04L 29/06(2006.01); H04L 29/08(2006.01); H04L 65/40(2022.01); H04W 28/20(2009.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models: IPC as above
Japanese utility models and applications for utility models: IPC as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS (KIPO internal) & keywords: 멀티 클라우드(multi-cloud), 하이퍼레저 패브릭 블록체인(hyperledger fabric blockchain), 클러스터(cluster), 트리(tree), 대역폭(bandwidth), 추정(estimation), 가우시안 모델(gaussian model)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	KR 10-2009-0053298 A (ICU RESEARCH AND INDUSTRIAL COOPERATION GROUP) 27 May 2009 (2009-05-27)	
X	See paragraphs [0007], [0034]-[0037], [0047], [0054]-[0055] and [0063].	1,14
Y		2-5,15-20
A		6-13
	KR 10-2021-0082890 A (SOGANG UNIVERSITY RESEARCH & BUSINESS DEVELOPMENT FOUNDATION) 06 July 2021 (2021-07-06)	
Y	See paragraphs [0027] and [0029].	2-5,15-20
	KR 10-2020-0062889 A (ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE) 04 June 2020 (2020-06-04)	
Y	See paragraphs [0010], [0049], [0054], [0066], [0088] and [0093]; and figures 3, 6a-6b and 8-9.	21-24,26-28
A		25,29-35

☑ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
21 September 2023	**22 September 2023**

Name and mailing address of the ISA/KR	Authorized officer
Korean Intellectual Property Office **Government Complex-Daejeon Building 4, 189 Cheongsa-ro, Seo-gu, Daejeon 35208**	
Facsimile No. **+82-42-481-8578**	Telephone No.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	KR 10-2019-0041784 A (FOURTH-LINK INC.) 23 April 2019 (2019-04-23) See paragraphs [0051] and [0072]-[0073]; and figure 4.	21-24,26-28
A	KANDA, Reiki et al. Estimation of Data Propagation Time on the Bitcoin Network. AINTEC `19: Proceedings of the 15th Asian Internet Engineering Conference. pp. 47–52, August 2019. See abstract.	1-20
A	KR 10-2022-0064744 A (TMAXENTERPRISE CO., LTD.) 19 May 2022 (2022-05-19) See paragraphs [0034]-[0039].	21-35

INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR2022/015119

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

The invention of group 1: claims 1-20 pertain to an effective bandwidth estimation method for transmission of block data in multicloud.
The invention of group 2: claims 21-35 pertain to a control message generation method and a tree information transmission method of a block transmission system for scalable Hyperledger Fabric blockchain in multicloud.

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☑ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/KR2022/015119

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
KR 10-2009-0053298 A	27 May 2009	KR 10-0932557 B1	17 December 2009
KR 10-2021-0082890 A	06 July 2021	KR 10-2541781 B1	08 June 2023
		US 11625260 B2	11 April 2023
		US 2021-0200570 A1	01 July 2021
KR 10-2020-0062889 A	04 June 2020	None	
KR 10-2019-0041784 A	23 April 2019	None	
KR 10-2022-0064744 A	19 May 2022	None	

국 제 조 사 보 고 서

국제출원번호
PCT/KR2022/015119

A. 발명이 속하는 기술분류(국제특허분류(IPC))

H04L 67/1074(2022.01)i; **H04L 43/0852**(2022.01)i; **H04L 43/0876**(2022.01)i; **H04L 67/1042**(2022.01)i; **H04L 67/1097**(2022.01)i; **H04L 12/18**(2006.01)i; **H04L 69/22**(2022.01)i

B. 조사된 분야

조사된 최소문헌(국제특허분류를 기재)

H04L 67/1074(2022.01); G06F 9/50(2006.01); H04B 7/26(2006.01); H04L 29/06(2006.01); H04L 29/08(2006.01); H04L 65/40(2022.01); H04W 28/20(2009.01)

조사된 기술분야에 속하는 최소문헌 이외의 문헌

한국등록실용신안공보 및 한국공개실용신안공보: 조사된 최소문헌란에 기재된 IPC
일본등록실용신안공보 및 일본공개실용신안공보: 조사된 최소문헌란에 기재된 IPC

국제조사에 이용된 전산 데이터베이스(데이터베이스의 명칭 및 검색어(해당하는 경우))

eKOMPASS(특허청 내부 검색시스템) & 키워드: 멀티 클라우드(multi-cloud), 하이퍼레저 패브릭 블록체인(hyperledger fabric blockchain), 클러스터(cluster), 트리(tree), 대역폭(bandwidth), 추정(estimation), 가우시안 모델(gaussian model)

C. 관련 문헌

카테고리*	인용문헌명 및 관련 구절(해당하는 경우)의 기재	관련 청구항
X	KR 10-2009-0053298 A (한국정보통신대학교 산학협력단) 2009.05.27 단락 [0007], [0034]-[0037], [0047], [0054]-[0055], [0063]	1,14
Y		2-5,15-20
A		6-13
Y	KR 10-2021-0082890 A (서강대학교산학협력단) 2021.07.06 단락 [0027], [0029]	2-5,15-20
Y	KR 10-2020-0062889 A (한국전자통신연구원) 2020.06.04 단락 [0010], [0049], [0054], [0066], [0088], [0093]; 및 도면 3, 6a-6b, 8-9	21-24,26-28
A		25,29-35
Y	KR 10-2019-0041784 A (주식회사 포스링크) 2019.04.23 단락 [0051], [0072]-[0073]; 및 도면 4	21-24,26-28

☑ 추가 문헌이 C(계속)에 기재되어 있습니다.

☑ 대응특허에 관한 별지를 참조하십시오.

* 인용된 문헌의 특별 카테고리:

"A" 특별히 관련이 없는 것으로 보이는 일반적인 기술수준을 정의한 문헌
"D" 본 국제출원에서 출원인이 인용한 문헌
"E" 국제출원일보다 빠른 출원일 또는 우선일을 가지나 국제출원일 이후에 공개된 선출원 또는 특허 문헌
"L" 우선권 주장에 의문을 제기하는 문헌 또는 다른 인용문헌의 공개일 또는 다른 특별한 이유(이유를 명시)를 밝히기 위하여 인용된 문헌
"O" 구두 개시, 사용, 전시 또는 기타 수단을 언급하고 있는 문헌
"P" 우선일 이후에 공개되었으나 국제출원일 이전에 공개된 문헌
"T" 국제출원일 또는 우선일 후에 공개된 문헌으로, 출원과 상충하지 않으며 발명의 기초가 되는 원리나 이론을 이해하기 위해 인용된 문헌
"X" 특별한 관련이 있는 문헌. 해당 문헌 하나만으로 청구된 발명의 신규성 또는 진보성이 없는 것으로 본다.
"Y" 특별한 관련이 있는 문헌. 해당 문헌이 하나 이상의 다른 문헌과 조합하는 경우로 그 조합이 당업자에게 자명한 경우 청구된 발명은 진보성이 없는 것으로 본다.
"&" 동일한 대응특허문헌에 속하는 문헌

국제조사의 실제 완료일	국제조사보고서 발송일
2023년09월21일(21.09.2023)	**2023년09월22일(22.09.2023)**

ISA/KR의 명칭 및 우편주소

대한민국 특허청
(35208) 대전광역시 서구 청사로 189, 4동 (둔산동, 정부대전청사)

팩스 번호 **+82-42-481-8578**

심사관

변성철

전화번호 **+82-42-481-8262**

국 제 조 사 보 고 서

국제출원번호
PCT/KR2022/015119

C. 관련 문헌

카테고리*	인용문헌명 및 관련 구절(해당하는 경우)의 기재	관련 청구항
A	REIKI KANDA 등, `Estimation of Data Propagation Time on the Bitcoin Network`, AINTEC `19: Proceedings of the 15th Asian Internet Engineering Conference, 페이지 47–52, 2019.08 요약	1-20
A	KR 10-2022-0064744 A (주식회사 티맥스엔터프라이즈) 2022.05.19 단락 [0034]-[0039]	21-35

국 제 조 사 보 고 서

국제출원번호
PCT/KR2022/015119

제3기재란 발명의 단일성이 결여된 경우의 의견(첫 번째 용지의 3의 계속)

본 국제조사기관은 본 국제출원에 다음과 같이 다수의 발명이 있다고 봅니다.

제1군 발명: 청구항 1-20은 멀티 클라우드에서의 블록 데이터 전송을 위한 유효 대역폭 추정 방법에 관한 것이고,
제2군 발명: 청구항 21-35는 멀티 클라우드의 확장 가능한 하이퍼레저 패블릭 블록체인을 위한 블록 전송 시스템의제어 메시지 생성 방법 및 트리 정보 전달 방법에 관한 것입니다.

1. ☐ 출원인이 모든 추가수수료를 기간 내에 납부하였으므로, 본 국제조사보고서는 모든 조사 가능한 청구항을 대상으로 합니다.

2. ☑ 추가수수료 납부를 요구하지 않고도 모든 조사 가능한 청구항을 조사할 수 있었으므로, 본 기관은 추가수수료 납부를 요구하지 아니하였습니다.

3. ☐ 출원인이 추가수수료의 일부만을 기간 내에 납부하였으므로, 본 국제조사보고서는 수수료가 납부된 청구항만을 대상으로 합니다. 구체적인 청구항은 아래와 같습니다.

4. ☐ 출원인이 기간 내에 추가수수료를 납부하지 아니하였습니다. 따라서 본 국제조사보고서는 청구범위에 처음 기재된 발명에 한정되어 있으며, 해당 청구항은 아래와 같습니다.

이의신청에 관한 기재

☐ 출원인의 이의신청 및 이의신청료 납부(해당하는 경우)와 함께 추가수수료가 납부되었습니다.

☐ 출원인의 이의신청과 함께 추가수수료가 납부되었으나 이의신청료가 보정요구서에 명시된 기간 내에 납부되지 아니하였습니다.

☐ 이의신청 없이 추가수수료가 납부되었습니다.

국 제 조 사 보 고 서
대응특허에 관한 정보

국제출원번호
PCT/KR2022/015119

국제조사보고서에서 인용된 특허문헌	공개일	대응특허문헌	공개일
KR 10-2009-0053298 A	2009/05/27	KR 10-0932557 B1	2009/12/17
KR 10-2021-0082890 A	2021/07/06	KR 10-2541781 B1	2023/06/08
		US 11625260 B2	2023/04/11
		US 2021-0200570 A1	2021/07/01
KR 10-2020-0062889 A	2020/06/04	없음	
KR 10-2019-0041784 A	2019/04/23	없음	
KR 10-2022-0064744 A	2022/05/19	없음	