관 인 생 략

출 원 번 호 통 지 서

출 원 일 자	2019.11.20
특 기 사 항	심사청구(무) 공개신청(무) 참조번호(IP190018NP1)
출 원 번 호	10-2019-0149666 (접수번호 1-1-2019-1193964-41)
출원인 명칭	포항공과대학교 산학협력단(2-2004-043336-1)
대리인 성명	특허법인 이상(9-2008-100021-0)
발명자 성명	홍원기 강창훈
발명의 명칭	합의 알고리즘 성능 검증을 위한 블록체인 네트워크 시뮬레이션 방법

특 허 청 장

<< 안내 >>

1. 귀하의 출원은 위와 같이 정상적으로 접수되었으며, 이후의 심사 진행상황은 출원번호를 통해 확인하실 수 있습니다.

2. 출원에 따른 수수료는 접수일로부터 다음날까지 동봉된 납입영수증에 성명, 납부자번호 등을 기재하여 가까운 우체국 또는 은행에 납부하여야 합니다.
※ 납부자번호 : 0131(기관코드) + 접수번호

3. 귀하의 주소, 연락처 등의 변경사항이 있을 경우, 즉시 [특허고객번호 정보변경(경정), 정정신고서]를 제출하여야 출원 이후의 각종 통지서를 정상적으로 받을 수 있습니다.
※ 특허로(patent.go.kr) 접속 > 민원서식다운로드 > 특허법 시행규칙 별지 제5호 서식

4. 특허(실용신안등록)출원은 명세서 또는 도면의 보정이 필요한 경우, 등록결정 이전 또는 의견서 제출기간 이내에 출원서에 최초로 첨부된 명세서 또는 도면에 기재된 사항의 범위 안에서 보정할 수 있습니다.

5. 외국으로 출원하고자 하는 경우 PCT 제도(특허·실용신안)나 마드리드 제도(상표)를 이용할 수 있습니다. 국내출원일을 외국에서 인정받고자 하는 경우에는 국내출원일로부터 일정한 기간 내에 외국에 출원하여야 우선권을 인정받을 수 있습니다.
※ 제도 안내 : http://www.kipo.go.kr-특허마당-PCT/마드리드
※ 우선권 인정기간 : 특허·실용신안은 12개월, 상표·디자인은 6개월 이내
※ 미국특허상표청의 선출원을 기초로 우리나라에 우선권주장출원 시, 선출원이 미공개상태이면, 우선일로부터 16개월 이내에 미국특허상표청에 [전자적교환허가서(PTO/SB/39)를 제출하거나 우리나라에 우선권 증명서류를 제출하여야 합니다.

【서지사항】

【서류명】 특허출원서

【참조번호】 IP190018NP1

【출원구분】 특허출원

【출원인】

【명칭】 포항공과대학교 산학협력단

【특허고객번호】 2-2004-043336-1

【대리인】

【명칭】 특허법인 이상

【대리인번호】 9-2008-100021-0

【지정된변리사】 이재관, 전호진, 김영아

【포괄위임등록번호】 2008-057306-7

【발명의 국문명칭】 합의 알고리즘 성능 검증을 위한 블록체인 네트워크 시뮬레이션 방법

【발명의 영문명칭】 Blockchain network simulation for verifying performance of consensus algorithms

【발명자】

【성명】 홍원기

【성명의 영문표기】 HONG, Won Ki

【주소】 경상북도 포항시 남구 지곡로 319, 328동 304호

【주소의 영문표기】 328-304, 319, Jigok-ro, Nam-gu, Pohang-si, Gyeongsangbuk-do

【발명자】

【성명】 강창훈

【성명의 영문표기】 KANG, Chang Hoon

【주민등록번호】 970507-1XXXXXX

【우편번호】 46957

【주소】 부산광역시 사상구 백양대로700번길 23, 101동 1003호

【출원언어】 국어

【이 발명을 지원한 국가연구개발사업】

【과제고유번호】 1711081028

【부처명】 과학기술정보통신부

【연구관리 전문기관】 한국연구재단

【연구사업명】 블록체인융합기술개발

【연구과제명】 블록체인의 트랜잭션 모니터링 및 분석 기술개발

【기여율】 1/1

【주관기관】 계명대학교 산학협력단

【연구기간】 2019.01.01 ~ 2019.12.31

【취지】 위와 같이 특허청장에게 제출합니다.

대리인 특허법인 이상 (서명 또는 인)

【수수료】

【출원료】 0 면 46,000 원

【가산출원료】 19 면 0 원

【우선권주장료】 0 건 0 원

【심사청구료】 0 항 0 원

【합계】 46,000 원

【감면사유】 전담조직(50%감면)[1]

【감면후 수수료】 23,000 원

【발명의 설명】

【발명의 명칭】

합의 알고리즘 성능 검증을 위한 블록체인 네트워크 시뮬레이션 방법{Blockchain network simulation for verifying performance of consensus algorithms}

【기술분야】

<0001> 본 발명은 사용자가 원하는 합의 알고리즘의 성능을 시뮬레이션을 바탕으로 한 실험을 통해 확인하는 방법에 관한 것이다.

【발명의 배경이 되는 기술】

<0002> 블록체인 (Blockchain)은 분산된 여러 개의 노드 (Node)들이 모두 동일한 데이터 원장을 저장 및 관리하도록 하여 누구도 데이터를 임의로 수정할 수 없으면서 이를 투명하게 관리하도록 하는 기술을 지칭한다. 다시 말해, 블록체인 기술은 Peer-to-peer 네트워크에서 분산된 원장을 이용해 두 당사자 간 거래를 할 때 은행처럼 신뢰할 수 있는 제 3의 기관의 필요성을 제거할 수 있는 기술이다.

<0003> 블록체인의 주요 특징은 탈중앙화 된 관리, 데이터의 투명성, 데이터의 불변성 등의 특징이 있다. 이러한 특징들을 이용해 기존 서비스들 중 중앙화된 방식으로 비롯된 문제점을 블록체인 적용을 통해 해결하려는 시도가 많이 이루어지고 있다.

<0004> 또한, 목적에 따라 특정 서비스의 특징들에 더욱 큰 비중을 두는 특화된 블록체인에 대한 연구 역시 많이 진행되고 있다. 이렇게 서로 다른 특장점들을 갖도

록 블록체인 네트워크의 동작 방식을 다르게 하는 것은 합의 알고리즘(Consensus Algorithm)이다.

<0005> 합의 알고리즘은 기본적으로 블록체인 네트워크 상의 많은 노드들이 모두 동일한 의사결정 과정을 통해 같은 데이터를 저장하고 관리하도록 하는 역할을 한다.

<0006> 예를 들어, 각각의 블록체인 네트워크마다 추구하는 방향이나 목적에 따라 그에 부합하는 합의 알고리즘을 채택하고 있다. 이에 따라 새로운 합의 알고리즘에 대한 연구과 개발 역시 활발히 이루어지고 있다.

<0007> ICO (Initial Coin Offering)와 함께 새로운 블록체인이 제안될 때, 연관된 백서 (White Paper)를 통하여 이 블록체인의 기반이 되는 새로운 합의 알고리즘에 대한 정보들을 제공한다. 성능이나 보안성 등의 측면에서 수학적 증명 과정을 통해 검증한 결과가 이에 해당한다.

<0008> 이때, 참여하는 노드들의 수나 네트워크의 구조 등에 따라서 해당 합의 알고리즘이 적용된 블록체인의 성능이 달라질 수 있는데, 한 번 실제 블록체인에 이를 적용하고 나면 이후 새로운 알고리즘으로 변경하는 것에 많은 비용이 소모되기 때문에 우선적으로 실험을 통한 검증을 거치는 것이 요구된다. 따라서 적은 비용으로도 정확도 높은 시뮬레이션을 통해 합의 알고리즘의 성능을 실험해볼 수 있는 방법이 필요하다.

<0009> 하지만 그것이 실제 블록체인 네트워크 환경에 적용되었을 때 어떻게 동작하는지를 확인하는 실험을 통한 검증은 많은 비용과 시간이 소모되는 등의 문제로 잘 이루어지지 않고 있다.

<0010> 또한, 블록체인 상에서는 매번 오직 하나의 동일한 블록만이 각 노드가 갖고 있는 사슬에 추가되어야 하는데, 이 과정에서 만약 여러 채굴자 (Miner)가 모두 동시에 블록 생성에 성공한 경우, 블록을 수신한 각 노드들이 어떤 블록을 추가할지 역시 결정해준다. 또한 네트워크 상의 전달 지연 (Propagation Delay)등의 이유로 같은 높이에 서로 다른 블록이 추가되어 포크 (fork)가 발생한 경우, 최장 사슬 (Longest Chain)을 선택하게 만들어서 노드들이 다시 모두 동일한 체인을 갖도록 하는 것도 합의 알고리즘의 역할 중 하나이다.

<0011> 현재의 합의 알고리즘 성능 검증은 대부분 수학적인 분석과 증명을 바탕으로 이루어지며 그 결과를 백서에 포함하여 배포하고 있다. 하지만 해당 알고리즘이 실험적으로 어느 정도의 성능을 보일지와 같은 내용은 제시되지 않는다.

<0012> 또한, 만약 실험을 진행하였더라도 굉장히 제한적인 환경을 구성하고 이를 활용했기 때문에 실제 환경에서도 같은 성능을 보일 것이라는 보장을 할 수 없는 실험이라는 문제가 있다. 이때 실험에서 제한적인 환경만을 구성한 이유는 실제와 비슷한 환경 구축을 위해서는 아주 많은 비용과 시간이 필요하기 때문이다. 분산된 지역에 수많은 노드들을 실제로 분포시키고 시뮬레이션을 진행할 수 있도록 준비하는 것뿐만 아니라 실험의 결과를 수집하고 시스템을 관리하는 것은 큰 어려움이 따른다.

<0013> 현재 시뮬레이션을 이용하여 블록체인 네트워크의 실험을 진행하는 기술은 몇몇 프로젝트들을 통해 연구되고 있다. 하지만 이는 본 발명이 목표로 하는 합의 알고리즘의 성능 검증과는 다른 목적을 갖고 있다는 문제가 있다. 해당 문제는 크

게 두 가지 경우로 나누어 볼 수 있다. 첫 번째 경우는 기존에 이미 존재하는 하나의 특정 합의 알고리즘에 종속적인 시뮬레이터이다. 따라서, 성능 검증을 위해 새로운 합의 알고리즘을 적용하는 것은 불가능하다. 두 번째 경우는 네트워크의 구조나 합의 알고리즘을 자유롭게 설정할 수 있도록 하였지만, 합의 알고리즘보다는 전체 시스템의 네트워킹 성능 측정에 초점을 두고 있다. 따라서 앞선 두 경우 모두 본 발명과는 다른 목적으로 기술이 개발되고 있어 합의 알고리즘의 성능 검증이라는 목적을 달성하기에는 무리가 있다.

【발명의 내용】

【해결하고자 하는 과제】

<0014> 상기와 같은 문제점을 해결하기 위한 본 발명의 목적은 하나의 물리적 머신 환경에서 합의 알고리즘의 성능을 측정 및 검증할 수 있는 시뮬레이터 모델을 구현하는 것이다.

<0015> 네트워크 에뮬레이터인 Mininet을 활용해 원하는 네트워크 구조를 생성하고, 각 호스트가 하나의 노드 역할을 하도록 해 블록체인 네트워크를 구성한다. 합의 알고리즘을 표현할 수 있는 여러 Parameter들을 제시하여, 각각의 Parameter 값을 설정하는 방법을 통해 앞서 구성한 블록체인 네트워크에 적용시킬 합의 알고리즘을 구현하는 것으로 시뮬레이터를 구성한다. 시간 흐름에 따른 시뮬레이션 과정의 데이터를 기록 및 분석하여 최종적으로 합의 알고리즘의 성능과 관련된 지표인 초당 최대 처리 거래 수 (Transactions Per Second, TPS), 블록 생성 주기, 각 거래가 완결될 때까지 필요한 시간 등의 값을 제공할 수 있도록 한다.

【과제의 해결 수단】

<0016> 상기 목적을 달성하기 위한 본 발명은 네트워크 에뮬레이터 중 하나인 Mininet을 이용해 블록체인 네트워크를 구성하고, 이를 원하는 방식으로 동작하기 위한 설정값들을 여러 Parameter를 통해 입력받는다. 이렇게 구성된 블록체인 네트워크에서 시간의 흐름에 따라 어떤 이벤트들이 발생하는지 시뮬레이션하고 성능과 관련된 여러 지표들의 측정값을 제공한다.

【발명의 효과】

<0017> 본 발명에서는 합의 알고리즘의 성능 측정 및 검증을 위해 많은 비용과 시간으로 제한적인 환경을 구성하여 실험하는 것이 아닌 적은 비용으로 빠르게 현실성 높은 환경에서 시뮬레이션을 진행한다. 하나의 물리적 머신에서 가상 블록체인 네트워크를 생성하고 해당 합의 알고리즘을 적용시켜 시뮬레이션 하기 때문이다. 기존 시뮬레이터들은 대부분 하나의 정해진 합의 알고리즘만 시뮬레이션 해볼 수 있었다면 본 발명은 하나의 프로그램으로 이용자가 원하는 합의 알고리즘을 자유롭게 구성하여 실험하는 것이 가능하다. 또한 본 발명으로 인해 추후 연구되는 합의 알고리즘들이 실험을 통한 성능 검증 과정까지 쉽게 거칠 수 있을 것이라 예상된다. 이에 따라 기존에 문제점이었던 실제 블록체인의 구축 후 예상했던 성능과는 다른 결과로 인해 새로운 합의 알고리즘을 고안하는 과정이 필요하게 되는 등의 상황은 줄어들 것이다.

【도면의 간단한 설명】

<0018> 도 1은 Mininet을 활용해 구성하는 네트워크가 동작하는 방식을 표현한 도면

이다.

도 2은 시뮬레이션이 진행되는 과정을 나타낸 도면이다.

【발명을 실시하기 위한 구체적인 내용】

<0019> 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세한 설명에 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. 각 도면을 설명하면서 유사한 참조부호를 유사한 구성요소에 대해 사용하였다.

<0020> 제1, 제2, A, B 등의 용어는 다양한 구성요소들을 설명하는 데 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.

<0021> 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 접

속되어" 있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다.

<0022> 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, "포함하다" 또는 "가지다" 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.

<0023> 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 가지는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

<0024> 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.

<0025>

<0026> 도 1은 Mininet을 활용해 구성하는 네트워크가 동작하는 방식을 표현한 도면이다.

<0027> 도 1을 참조하면, 이번 발명에서 제안하는 시뮬레이터 모델은 도 1의 방식으로 가상의 블록체인 네트워크를 구성할 수 있다.

<0028> 하나의 머신 위에서 수많은 노드들이 연결된 블록체인 네트워크 구조의 구성을 위해 본 발명은 네트워크 에뮬레이터 Mininet을 활용한다.

<0029> 도 1에서처럼 Mininet에 네트워크 구조와 관련된 설정들을 적용하면 본인이 원하는 네트워크 환경을 구성할 수 있다. 하나의 노드 역할을 하게 될 호스트들의 수, 스위치들과 함께 연결된 전체적인 네트워크 구조 역시 설정이 가능하다. 이때 본 발명의 이용자들이 Mininet의 사용법까지 익혀야 하는 부담이 생기지 않도록 이용자로부터 간단한 옵션들을 입력 받는 것을 통해 이를 바탕으로 시뮬레이터가 직접 네트워크를 구성할 수 있도록 한다. 또한 일반적인 경우의 네트워크 구조 환경 설정을 미리 제공하여 사용할 수 있도록 한다.

<0030> [표 1]의 Parameter들을 입력받아 합의 알고리즘을 적용시킨 후 시뮬레이션을 시작한다. 여기서, [표 1]은 시뮬레이터 설정을 위한 Parameter들의 종류를 작성한 표이다.

<0031>

<0032> 【표 1】

	네트워크	노드	합의 알고리즘
Parameters	• 총 노드의 수 • 대역폭 • 평균 전달 지연 • 평균 거래 크기 • 블록 크기 • 거래 생성 주기	• 필요 변수 종류 • 각 변수의 해당 값 • 채굴자 여부	• 다음 블록 선정 기준 변수 • 각 변수의 가중치 • 확률적 선택 여부 • 보상 (사용하지 않는 경우, -1)

<0033>

<0034> 합의 알고리즘들은 각각 서로 다른 특징들과 동작 과정을 갖고 있다. 따라서 모든 합의 알고리즘들을 표현하여 적용 가능하도록 만들기 위해서는 이들을 모두 표현할 수 있는 일반적인 특징들을 Parameter들로 추출하는 것이 필요하다. 이용자들은 이 Parameter들을 실험하고자 하는 합의 알고리즘에 알맞게 설정하는 과정을 통해 블록체인 네트워크 상에 합의 알고리즘을 적용시킨다. [표 1]에서 해당 Parameter들을 제시하고 있다.

<0035> 이때 전체 프로그램은 파이썬 3을 프로그래밍 언어로 사용하고, Mininet의 Python API를 사용한다. Parameter를 입력받을 때에는 json형식의 config 파일로 저장된 값을 읽어와 Parsing하여 사용한다.

<0036> 먼저 네트워크 구성과 관련된 파라미터들은 총 여섯 개이다. Mininet을 통해 구성하려는 네트워크의 구조에 대한 정보를 갖는 값들과, 블록체인 네트워크 동작에 있어서 필요한 정보들을 갖는 값들로 나눌 수 있다. 먼저 첫 번째 경우에 대해서는 네트워크 내의 총 노드의 개수, 네트워크 상 각 링크의 대역폭과 전달 지연 값이 이에 해당된다. 이때 모든 링크들이 같은 대역폭과 전달 지연 값을 가진다 가

정하고 구현한다. 나머지 파라미터들은 앞서 말했듯이 블록체인 네트워크에서 필요한 설정 정보들이다. 우선 조금씩 다르게 발생되는 트랜젝션의 크기를 모두 다르게 적용시키는 것에는 어려움이 있기 때문에, 평균적인 트랜젝션의 크기를 입력 및 적용시켰다. 그 뒤에는 각 블록체인에 따라 고정되어 있는 최대 블록의 크기에 대한 정보를 받아오도록 하였다. 마지막으로 트랜젝션을 발생시킬 주기를 입력할 수 있도록 하였다. 이 시뮬레이션이 동작하는데에 있어서 가장 기초가 되는 이벤트는 트랜젝션들의 발생이다. 전체 네트워크 상에서 어느 정도의 주기로 트랜젝션을 발생시키는지 입력한다면 시뮬레이션에서는 네트워크 위 노드들 중 하나를 주기마다 하나씩 무작위로 선정하여 해당 노드에서 트랜젝션이 발생하고, 이것이 다른 주변 모든 노드들로 전파되는 것을 시뮬레이션 할 수 있다.

<0037> 다음으로 각 노드 구성에 대한 설정이 필요하다. 우선 사용자 입력에 관계없이 노드들에게 순차적으로 id를 부여해준다. 이에 추가적으로 필요한 변수들의 정보와 그 초기값을 입력 받는다. 마지막으로 해당 노드가 채굴자인지 여부를 결정하는 값을 입력 받아 해당 노드가 채굴에 참여할지 여부를 결정한다.

<0038> 합의 알고리즘을 표현하는 Parameter들로는 네 개의 값이 있다.

<0039> 먼저 다음번에 새롭게 추가될 블록을 선택하는 것에 있어 기준이 되는 변수들에 대한 정보를 받는다. 이는 앞서 입력 받은 노드가 가지는 변수들 중 선택한다. 따라서 블록이 만들어지면 그것을 생산한 노드의 해당 기준 값에 따라 알고리즘에 맞게 합의를 이루어 하나의 블록이 선택되도록 한다. 다음으로 각 변수에 대한 가중치를 입력 받는다. 기준이 되는 값이 여러 개일 경우 각각의 가중치를 곱하

여 우선 순위를 결정하게 되고, 만약 기준값이 하나일 경우에도 해당 값이 크거나 작음에 따라 우선순위가 달라질 수도 있도록 하였다. 이외에도 앞서 설명한 기준값에 가중치를 곱한 값에 따라 단순 대소 비교를 통해 하나를 채택할 것인지, 아니면 값에 크기에 비례하여 더욱 높은 확률로 채택될 수 있는 시스템을 사용할 것인지 선택하는 Parameter이다.

<0040> 특히나 많은 계산을 필요로 하는 합의 알고리즘의 경우에는 많은 노드들의 계산을 모두 한 머신에서 처리하도록 하기 힘들기 때문에 이 같은 구현 방식이 필요하다. 마지막으로 만약 특정 노드가 채굴한 블록이 채택되었을 경우, 그 결과에 의해 다음 채굴자 선택의 기준이 되는 변수값이 바뀌어야 한다면 Reward를 통해 해당 값을 변경시킬 수 있도록 한다. 반대로 Reward가 없도록 하는 설정을 통해 이전의 채굴 성공 여부가 앞으로의 채굴 성공 여부에 영향을 주지 않는 독립적인 사건이 되도록 할 수도 있다.

<0041> 다시 말해, 부족한 컴퓨팅 자원을 효율적으로 사용하기 위해 필요한 경우 반복적인 복잡한 연산대신 확률에 따라 다음 블록 생성에 성공한 채굴자를 선정할 수 있게 한다. 적용시키려는 합의 알고리즘에서의 기준에 따라 각 노드들이 채굴에 성공할 수 있는 확률을 다르게 설정해주고 실험 중에도 상황에 따라 유동적으로 변동될 수 있도록 한다.

<0042>

<0043> [표 2]와 [표 3]은 위에서 설명한 방법을 통해 각각 작업증명과 지분증명 합의 알고리즘을 구현하는 예시를 나타낸 것이다. 다시 말하면, [표 2]와 [표 3]은

각각 작업증명과 지분증명을 시뮬레이터에 구현하려고 할 때의 Parameter 설정 예시이다.

<0044> [표 2]는 합의 알고리즘으로 작업증명 (Proof of Work, PoW)를 사용할 경우의 입력값 형태를 나타낸 표이다.

<0045>

<0046> 【표 2】

PoW	네트워크	노드	합의 알고리즘
Parameters	• 총 노드의 수 • 대역폭 • 평균 전달 지연 • 평균 거래 크기 • 블록 크기 • 거래 생성 주기	• [id, Hash Power(>0), Miner?]	• 다음 블록 선정 기준 변수 ➢ Hash Power • 각 변수의 가중치 ➢ (1) • 확률적 선택 여부 ➢ True • 보상 (사용하지 않는 경우, -1) ➢ -1

<0047>

<0048> 작업증명의 경우 각 노드의 Hash Rate를 기준으로, 지분증명의 경우 각 노드의 보유 지분(Stake)을 기준으로 확률을 부여한다.

<0049> [표 2]의 작업증명 합의 알고리즘에서 블록을 채굴하는 과정은 블록 헤더의 해시값이 특정 조건을 만족하는 Nonce값을 반복적인 계산을 통해 빠르게 찾는 것이다. 따라서 각각의 노드들의 Hash Power가 클수록 더욱 값을 빠르게 찾을 확률이 높아진다. 제로 모든 노드들이 해당 값을 찾기 위해 해시값을 반복적으로 계산하는 과정에서 많은 컴퓨팅 파워를 사용한다.

<0050> 하지만 본 발명의 시뮬레이터는 단 한 대의 머신, 즉 제한되어 있는 자원을

활용하여 많은 노드들이 모든 계산을 이루어내야 한다. 따라서 이 부분에서 앞서 설명하였던 기준치를 확률적으로 고려하는 방법을 사용할 수 있다. 각 노드의 Hash Power에 따라 이에 비례하여 확률적으로 해당 노드가 생성한 블록이 다음 블록으로 채택되도록 하는 방법을 사용했다.

<0051>

<0052> 마지막으로 작업증명 알고리즘은 특정 노드가 생성한 블록이 채택되더라도 그로부터 얻는 리워드가 합의 알고리즘에서 우선 순위에 영향을 주는 기준 값에 아무런 변화를 주지 않기 때문에 Reward 값을 -1로 설정한다.

<0053>

<0054> [표 3]은 합의 알고리즘으로 지분증명 (Proof of Stake, PoS)를 사용할 경우의 입력값 형태를 나타낸 표이다.

<0055>

<0056> 【표 3】

PoS	네트워크	노드	합의 알고리즘
Parameters	• 총 노드의 수 • 대역폭 • 평균 전달 지연 • 평균 거래 크기 • 블록 크기 • 거래 생성 주기	• [id, Owned Stake(>=0), Miner?]	• 다음 블록 선정 기준 변수 ➢ Owned Stake • 각 변수의 가중치 ➢ (1) • 확률적 선택 여부 ➢ True • 보상 (사용하지 않는 경우, -1) ➢ 10

<0057>

<0058> [표 3]에서 지분증명 합의 알고리즘의 예시를 앞서 설명한 작업증명과 비교

해 보았을 때, 이 경우 각 노드가 갖는 변수가 Hash Power에서 소유한 지분(Owned Stake)의 크기로 바뀌었고, 다음 블록 선택의 기준이 되는 변수 역시 동일하게 바뀌었다. 마지막으로 이 경우에는 Reward가 발생한다. 특정 노드가 생산한 블록이 채택되는 경우 해당 노드에게 리워드가 지급되어 갖고 있는 자산이 더욱 증가하게 되는데, 이때 그 값이 합의 알고리즘에서 우선 순위를 판단하는 기준이 된다. 따라서 바로 해당 Reward만큼 소유한 자산 (Owned Stake) 값을 증가시켜준다.

<0059>

<0060> 도 2는 시뮬레이션이 진행되는 과정을 나타낸 도면이다.

<0061> 도 2를 참조하면, 시뮬레이터는 이용자가 지정한 주기의 시간마다 임의의 노드에서 트랜젝션을 생성시킨다. 그렇게 생성된 트랜젝션들은 다른 노드들로 전송되고 각 노드들의 메모리풀에 저장된다. 이와 동시에 채굴자인 각 노드들은 메모리풀에 저장되어 있는 트랜젝션들 중 일부를 선택하여 블록으로 가공한다. 이렇게 생성된 블록역시 각 노드들에게 전파되고, 각 노드에서는 받은 블록들 중 어떤 것을 본인의 체인에 추가할지, 지정된 합의 알고리즘에 의해 선택해 자신의 체인을 연장한다.

<0062> 본 발명에서 성능 검증에 필요한 데이터 측정은 TPS, 블록 생성 주기, 각 거래가 완결될 때까지 필요한 시간을 측정한다. 시뮬레이션 하는 동안 지속적으로 해당 값들을 측정하여 로그 파일에 기록한다. 이외에도 필요한 데이터가 있다면 간단한 프로그래밍을 통해 얻을 수 있도록 수정할 수 있다.

<0063>

<0064> 본 발명의 실시예에 따른 방법의 동작은 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 프로그램 또는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의해 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어 분산 방식으로 컴퓨터로 읽을 수 있는 프로그램 또는 코드가 저장되고 실행될 수 있다.

<0065> 또한, 컴퓨터가 읽을 수 있는 기록매체는 롬(rom), 램(ram), 플래시 메모리(flash memory) 등과 같이 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치를 포함할 수 있다. 프로그램 명령은 컴파일러(compiler)에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터(interpreter) 등을 사용해서 컴퓨터에 의해 실행될 수 있는 고급 언어 코드를 포함할 수 있다.

<0066> 본 발명의 일부 측면들은 장치의 문맥에서 설명되었으나, 그것은 상응하는 방법에 따른 설명 또한 나타낼 수 있고, 여기서 블록 또는 장치는 방법 단계 또는 방법 단계의 특징에 상응한다. 유사하게, 방법의 문맥에서 설명된 측면들은 또한 상응하는 블록 또는 아이템 또는 상응하는 장치의 특징으로 나타낼 수 있다. 방법 단계들의 몇몇 또는 전부는 예를 들어, 마이크로프로세서, 프로그램 가능한 컴퓨터 또는 전자 회로와 같은 하드웨어 장치에 의해(또는 이용하여) 수행될 수 있다. 몇몇의 실시예에서, 가장 중요한 방법 단계들의 하나 이상은 이와 같은 장치에 의해 수행될 수 있다.

<0067> 실시예들에서, 프로그램 가능한 로직 장치(예를 들어, 필드 프로그머블 게이

트 어레이)가 여기서 설명된 방법들의 기능의 일부 또는 전부를 수행하기 위해 사용될 수 있다. 실시예들에서, 필드 프로그머블 게이트 어레이는 여기서 설명된 방법들 중 하나를 수행하기 위한 마이크로프로세서와 함께 작동할 수 있다. 일반적으로, 방법들은 어떤 하드웨어 장치에 의해 수행되는 것이 바람직하다.

<0068> 이상 본 발명의 바람직한 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】

네트워크 에뮬레이터로 네트워크를 구성하고, 이것에 원하는 동작 방식의 블록체인 시스템을 구현하여 이를 시뮬레이션 하는 시스템.

【청구항 2】

청구항 1에 있어서,

상기 시뮬레이션 결과로 초당 최대 처리 거래 수, 블록 생성 주기, 각 거래가 완결될 때까지 필요한 시간 등 적용된 합의 알고리즘의 성능을 보여주는 지표가 측정되는 시스템.

【청구항 3】

청구항 1에 있어서,

사용자로부터 입력받은 설정값들을 바탕으로 알아서 네트워크 구성 및 시뮬레이션을 수행하는 시스템.

【청구항 4】

청구항 3에 있어서,

JSON 형식 파일에 미리 구성된 설정 변수들의 값을 입력하는 방법으로 네트워크 및 블록체인 시뮬레이션 설정을 하는 시스템.

【요약서】

【요약】

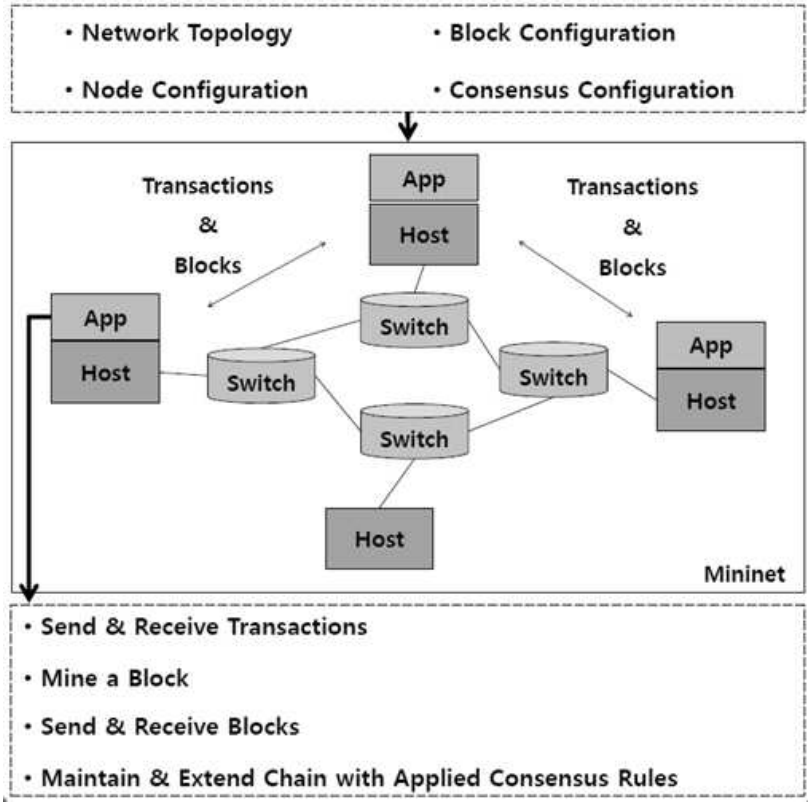
본 발명은 블록체인 네트워크에서 의사결정 과정을 담당하는 합의 알고리즘의 성능을 측정 및 분석하는 데에 있어 기존에 비해 적은 비용과 시간으로 더욱 현실성 높은 시뮬레이션을 실시할 수 있는 모델을 개발하고자 한다. 제안하는 방법에서는 네트워크 에뮬레이터인 Mininet을 통해 간편하게 가상 네트워크를 한 물리적 머신 상에 구현하고, 합의 알고리즘을 표현할 수 있는 Parameter들을 이용자가 설정해주어 이를 쉽게 블록체인 네트워크 상에 적용시킬 수 있다. 본 발명에서는 해당 방법을 통해 구동하는 시뮬레이터의 원리와 작동 과정에 대해서 자세히 소개하고 있다.

【대표도】

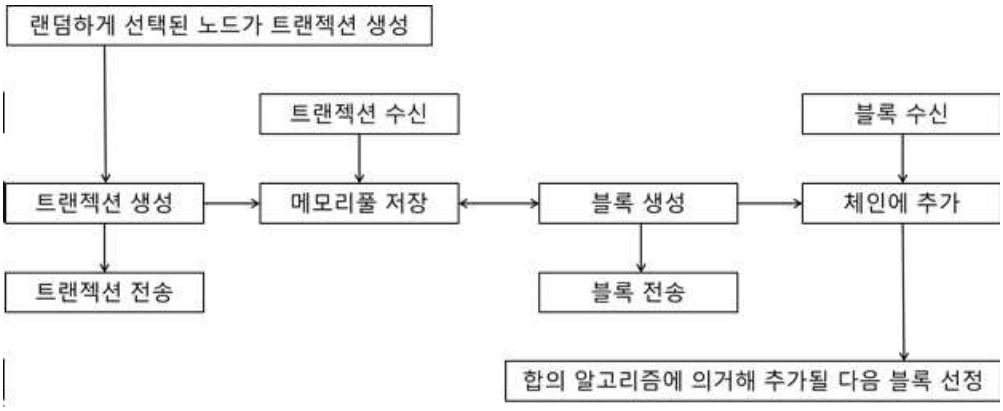
도 1

【도면】

【도 1】


• Network Topology
• Block Configuration
• Node Configuration
• Consensus Configuration
Transactions & Blocks
App
Host
Transactions & Blocks
App
Host
Switch
Switch
Switch
Switch
App
Host
Host
Mininet
• Send & Receive Transactions
• Mine a Block
• Send & Receive Blocks
• Maintain & Extend Chain with Applied Consensus Rules

【도 2】


랜덤하게 선택된 노드가 트랜젝션 생성
트랜젝션 수신
블록 수신
트랜젝션 생성
메모리풀 저장
블록 생성
체인에 추가
트랜젝션 전송
블록 전송
합의 알고리즘에 의거해 추가될 다음 블록 선정