



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0175574
(43) 공개일자 2024년12월20일

(51) 국제특허분류(Int. Cl.)
H04L 41/122 (2022.01) H04L 49/60 (2022.01)
H04L 9/00 (2022.01) H04L 9/32 (2006.01)
(52) CPC특허분류
H04L 41/122 (2022.05)
H04L 49/60 (2022.05)
(21) 출원번호 10-2023-0075677
(22) 출원일자 2023년06월13일
심사청구일자 2023년06월13일

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
김항남
서울특별시 서초구 서운로 104, 205동 503호 (서초동, 래미안 서초 에스티지S)
유태훈
서울특별시 동대문구 안암로16길 18, 정보빌 206호
주현태
서울특별시 동대문구 고산자로34가길 35 (용두동) 103호
(74) 대리인
윤귀상

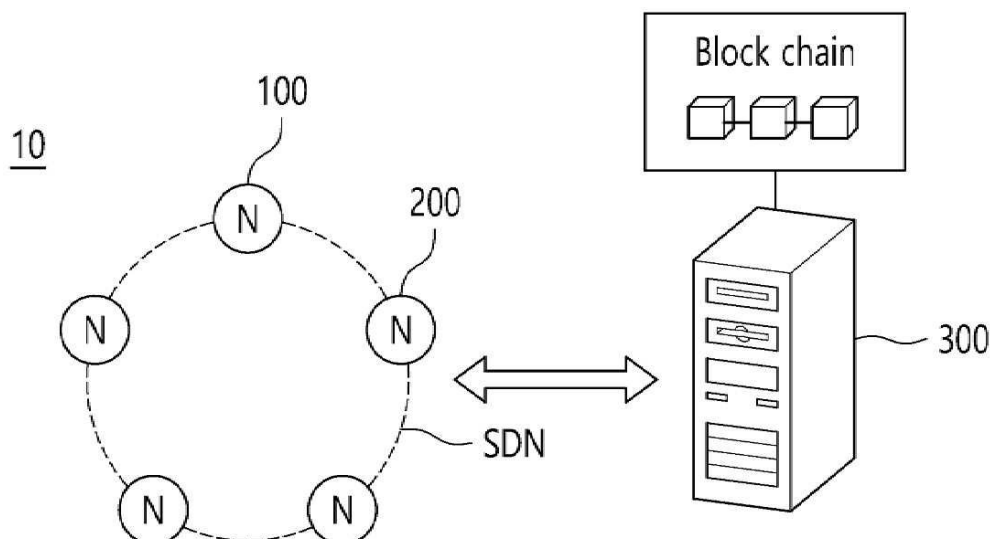
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 블록체인 기반의 SDN 참여자 관리 시스템 및 방법

(57) 요약

본 발명은 블록체인 기반의 SDN 참여자 관리 시스템에 관한 것으로, 참여 시간을 갱신하기 위한 갱신 트랜잭션을 생성하여 SDN의 참여를 요청하는 제1 노드; 갱신 트랜잭션에 대한 검증을 수행하는 제2 노드; 및 검증 결과에 따라 상기 제1 노드에게 갱신된 참여 시간을 부여하고, 블록을 생성하여 블록체인에 추가하는 SDN 컨트롤러를 포함한다. 이를 통해 SDN에서 새로운 규칙을 배포하거나 네트워크를 구성하는 노드가 새롭게 네트워크에 참여하는 과정에서 유효성 검사 과정을 보장하여 악의적인 노드의 네트워크 가입을 제한할 수 있다. 그리고 SDN 네트워크에 소속된 노드들의 지속적인 네트워크 참여 여부를 결정하여 SDN 네트워크 연결 정보를 안전하게 관리하는 것은 물론, 전체 시스템의 신뢰성을 보장하여 안정적으로 유지할 수 있다.

대표도 - 도1



(52) CPC특허분류

H04L 9/3297 (2013.01)

H04L 9/50 (2022.05)

이 발명을 지원한 국가연구개발사업

과제고유번호	1415186969
과제번호	20204010600220
부처명	산업통상자원부
과제관리(전문)기관명	한국에너지기술평가원
연구사업명	에너지인력양성사업(예특)
연구과제명	에너지거래 플랫폼 융합인력양성
기 여 율	1/1
과제수행기관명	건국대학교산학협력단
연구기간	2023.01.01 ~ 2023.12.31

명세서

청구범위

청구항 1

복수의 노드를 포함하여 구성되는 SDN(Software Defined Networking)에서 블록체인에 기반하여 상기 노드의 참여를 관리하는 블록체인 기반의 SDN 참여자 관리 시스템으로서,

상기 SDN에 참여 가능하도록 허용된 참여 시간을 갱신하기 위한 갱신 트랜잭션을 생성하여 상기 SDN의 참여를 요청하는 제1 노드;

상기 갱신 트랜잭션에 대한 검증을 수행하는 제2 노드; 및

상기 복수의 노드와 연결되어 상기 복수의 노드를 관리 및 제어하는 SDN 컨트롤러로서, 상기 검증 결과에 따라 상기 제1 노드에게 갱신된 참여 시간을 부여하고, 블록을 생성하여 상기 블록을 블록체인에 추가하는 상기 SDN 컨트롤러를 포함하는, 블록체인 기반의 SDN 참여자 관리 시스템.

청구항 2

제1항에 있어서,

상기 SDN 컨트롤러는,

상기 SDN을 최초로 구성할 때 상기 복수의 노드 각각에게 참여 시간 및 참여 요청 가능 시간을 부여하는, 블록체인 기반의 SDN 참여자 관리 시스템.

청구항 3

제2항에 있어서,

상기 제1 노드는,

상기 복수의 노드 중에서 상기 참여 시간을 상기 블록이 생성되는 주기 미만으로 부여받은 요청 노드이고,

상기 제2 노드는,

상기 복수의 노드 중에서 상기 참여 시간을 상기 블록이 생성되는 주기 이상으로 부여받은 검증 노드인, 블록체인 기반의 SDN 참여자 관리 시스템.

청구항 4

제1항에 있어서,

상기 갱신 트랜잭션은,

트랜잭션의 인덱스, 현재 허용된 참여 시간, 상기 제1 노드와 현재 연결된 인접 노드의 수, 갱신할 참여 시간, 상기 제1 노드와 연결될 인접 노드의 수 중 적어도 하나를 포함하고,

상기 제1 노드는,

상기 생성된 갱신 트랜잭션을 네트워크 상에 전파하는, 블록체인 기반의 SDN 참여자 관리 시스템.

청구항 5

제4항에 있어서,

상기 제2 노드는,

상기 복수의 노드 중에서 상기 갱신 트랜잭션을 수신한 노드로서, 상기 갱신 트랜잭션을 사전에 마련되는 참여 시간 합의 알고리즘에 기초하여 검증하고, 검증이 완료된 갱신 트랜잭션을 댄풀(Mempool)로 전송하는, 블록체인 기반의 SDN 참여자 관리 시스템.

청구항 6

제5항에 있어서,

상기 제2 노드는,

상기 참여 시간 합의 알고리즘에 따라 현재 타임스탬프가 상기 현재 허용된 참여 시간과 상기 갱신할 참여 시간 사이인지 확인하여 상기 갱신 트랜잭션을 검증하는, 블록체인 기반의 SDN 참여자 관리 시스템.

청구항 7

제6항에 있어서,

상기 제2 노드는,

상기 참여 시간 합의 알고리즘에 따라 상기 현재 연결된 인접 노드의 수와 상기 연결될 인접 노드의 수에 기초한 상기 제1 노드의 네트워크 연결 상태를 확인하여 상기 갱신 트랜잭션을 검증하는, 블록체인 기반의 SDN 참여자 관리 시스템.

청구항 8

제7항에 있어서,

상기 제2 노드는,

상기 갱신 트랜잭션을 검증한 결과에 대응하는 비율이 사전에 설정된 임계 비율을 초과하면 해당 갱신 트랜잭션이 유효한 것으로 판단하고, 유효한 갱신 트랜잭션을 상기 맴폴로 전송하는, 블록체인 기반의 SDN 참여자 관리 시스템.

청구항 9

제5항에 있어서,

상기 SDN 컨트롤러는,

일정 주기마다 상기 맴폴에 저장된 상기 검증이 완료된 갱신 트랜잭션을 포함하는 블록을 생성하고, 생성된 블록을 블록체인에 추가하는, 블록체인 기반의 SDN 참여자 관리 시스템.

청구항 10

복수의 노드를 포함하여 구성되는 SDN에서 상기 복수의 노드와 연결되고 블록체인에 기반하여 상기 복수의 노드를 관리 및 제어하는 SDN 컨트롤러를 포함하는 블록체인 기반의 SDN 참여자 관리 시스템에서의 블록체인 기반의 SDN 참여자 관리 방법으로서,

제1 노드가, 상기 SDN에 참여 가능하도록 허용된 참여 시간을 갱신하기 위한 갱신 트랜잭션을 생성하고, 상기 갱신 트랜잭션을 네트워크 상에 전파하여 상기 SDN의 참여를 요청하는 단계;

제2 노드가, 상기 갱신 트랜잭션에 대한 검증을 수행하는 단계; 및

상기 SDN 컨트롤러가, 상기 검증 결과에 따라 상기 제1 노드에게 갱신된 참여 시간을 부여하고, 블록을 생성하여 상기 블록을 블록체인에 추가하는 단계를 포함하는, 블록체인 기반의 SDN 참여자 관리 방법.

발명의 설명**기술 분야**

[0001] 본 발명은 SDN(Software Defined Networking)에서 안정적인 참여자 관리를 보장할 수 있는 블록체인 기반의 SDN 참여자 관리 시스템 및 방법에 관한 것이다.

배경 기술

[0002] SDN(Software Defined Networking)은 네트워크 구조를 소프트웨어로 제어하고 관리하는 기술이다. 구체적으로 SDN은 네트워크 제어의 간소화를 위해 네트워킹 리소스를 가상화하고, 구조적으로 네트워크 제어 영역(Control

Plane)과 데이터 포워딩 영역(Data Plane)을 분리하여, 전체 네트워크를 프로그래밍이 가능하도록 추상화하여 네트워크를 관리하는 기술이다.

[0003] 여기서 제어 영역은 네트워크를 제어하는 논리적인 부분으로써 네트워크 관리자가 네트워크를 제어할 수 있는 인터페이스를 제공하는 역할을 수행한다. 한편 데이터 포워딩 영역은 실제 데이터 전송을 담당하는 부분으로써 스위치, 라우터 등 네트워크 장비들이 데이터를 주고받는 역할을 수행한다.

[0004] 이러한 SDN의 경우 중앙 집중화된 제어 영역을 가질 수 있기 때문에 해당 제어 영역에 장애가 발생하면 전체 네트워크에 장애가 발생하는 단일 장애 지점(single point failure) 문제를 가질 수 있어 보안상 중요한 문제점을 갖는다.

[0005] 또한 현재까지 SDN의 표준화가 이루어지지 않았기 때문에 다양한 네트워크 제조업체와 개발자 간의 다양한 SDN 구현 방식과 프로토콜에 대한 호환성 문제가 있을 수 있다.

[0006] 따라서 기존의 네트워크 장비들을 대체하거나 업그레이드하는 과정에서 복잡한 소프트웨어 관리 작업이 필요하다면 이는 오히려 네트워크 관리를 복잡하게 하는 문제를 야기할 수 있다.

선행기술문헌

특허문헌

[0007] (특허문헌 0001) 한국등록특허공보 제10-1481337호

발명의 내용

해결하려는 과제

[0008] 본 발명은 상기와 같은 문제를 해결하기 위해 안출된 것으로, 본 발명의 목적은 SDN에서 새로운 규칙을 배포하거나 네트워크를 구성하는 노드가 새롭게 네트워크에 참여하는 과정에서 유효성 검사 과정을 보장하여 악의적인 노드의 네트워크 가입을 제한할 수 있는 블록체인 기반의 SDN 참여자 관리 시스템 및 방법을 제공하는 것이다.

[0009] 본 발명의 또 다른 목적은 SDN 네트워크에 소속된 노드들의 지속적인 네트워크 참여 여부를 결정함으로써 SDN 네트워크 연결 정보를 안전하게 관리하는 것은 물론, 전체 시스템의 신뢰성을 보장하여 안정적으로 유지할 수 있는 블록체인 기반의 SDN 참여자 관리 시스템 및 방법을 제공하는 것이다.

과제의 해결 수단

[0010] 상기 목적을 달성하기 위한 본 발명의 일 실시예에 따른 복수의 노드를 포함하여 구성되는 SDN(Software Defined Networking)에서 블록체인에 기반하여 상기 노드의 참여를 관리하는 블록체인 기반의 SDN 참여자 관리 시스템은, 상기 SDN에 참여 가능하도록 허용된 참여 시간을 갱신하기 위한 갱신 트랜잭션을 생성하여 상기 SDN의 참여를 요청하는 제1 노드; 상기 갱신 트랜잭션에 대한 검증을 수행하는 제2 노드; 및 상기 복수의 노드와 연결되어 상기 복수의 노드를 관리 및 제어하는 SDN 컨트롤러로서, 상기 검증 결과에 따라 상기 제1 노드에게 갱신된 참여 시간을 부여하고, 블록을 생성하여 상기 블록을 블록체인에 추가하는 상기 SDN 컨트롤러를 포함한다.

[0011] 그리고 상기 SDN 컨트롤러는, 상기 SDN을 최초로 구성할 때 상기 복수의 노드 각각에게 최초 참여 시간 및 참여 요청 가능 시간을 부여할 수 있다.

[0012] 또한 상기 제1 노드는, 상기 복수의 노드 중에서 상기 참여 시간을 상기 블록이 생성되는 주기 미만으로 부여받은 요청 노드이고, 상기 제2 노드는, 상기 복수의 노드 중에서 상기 참여 시간을 상기 블록이 생성되는 주기 이상으로 부여받은 검증 노드일 수 있다.

[0013] 그리고 상기 갱신 트랜잭션은, 트랜잭션의 인덱스, 현재 허용된 참여 시간, 상기 제1 노드와 현재 연결된 인접 노드의 수, 갱신할 참여 시간, 상기 제1 노드와 연결될 인접 노드의 수 중 적어도 하나를 포함하고, 상기 제1 노드는, 상기 생성된 갱신 트랜잭션을 네트워크 상에 전파할 수 있다.

[0014] 또한 상기 제2 노드는, 상기 복수의 노드 중에서 상기 갱신 트랜잭션을 수신한 노드로서, 상기 갱신 트랜잭션을

사전에 마련되는 참여 시간 합의 알고리즘에 기초하여 검증하고, 검증이 완료된 갱신 트랜잭션을 댄폴(Mempool)로 전송할 수 있다.

[0015] 그리고 상기 제2 노드는, 상기 참여 시간 합의 알고리즘에 따라 현재 타임스탬프(Time stamp)가 상기 현재 허용된 참여 시간과 상기 갱신할 참여 시간 사이인지 확인하여 상기 갱신 트랜잭션을 검증할 수 있다.

[0016] 또한 상기 제2 노드는, 상기 참여 시간 합의 알고리즘에 따라 상기 현재 연결된 인접 노드의 수와 상기 연결된 인접 노드의 수에 기초한 상기 제1 노드의 네트워크 연결 상태를 확인하여 상기 갱신 트랜잭션을 검증할 수 있다.

[0017] 그리고 상기 제2 노드는, 상기 갱신 트랜잭션을 검증한 결과에 대응하는 비율이 사전에 설정된 임계 비율을 초과하면 해당 갱신 트랜잭션이 유효한 것으로 판단하고, 유효한 갱신 트랜잭션을 상기 댄폴로 전송할 수 있다.

[0018] 또한 상기 SDN 컨트롤러는, 일정 주기마다 상기 댄폴에 저장된 상기 검증이 완료된 갱신 트랜잭션을 포함하는 블록을 생성하고, 생성된 블록을 블록체인에 추가할 수 있다.

[0019] 한편 상기 목적을 달성하기 위한 본 발명의 일 실시예에 따른 복수의 노드를 포함하여 구성되는 SDN에서 상기 복수의 노드와 연결되고 블록체인에 기반하여 상기 복수의 노드를 관리 및 제어하는 SDN 컨트롤러를 포함하는 블록체인 기반의 SDN 참여자 관리 시스템에서의 블록체인 기반의 SDN 참여자 관리 방법은, 제1 노드가, 상기 SDN에 참여 가능하도록 허용된 참여 시간을 갱신하기 위한 갱신 트랜잭션을 생성하고, 상기 갱신 트랜잭션을 네트워크 상에 전파하여 상기 SDN의 참여를 요청하는 단계; 제2 노드가, 상기 갱신 트랜잭션에 대한 검증을 수행하는 단계; 및 상기 SDN 컨트롤러가, 상기 검증 결과에 따라 상기 제1 노드에게 갱신된 참여 시간을 부여하고, 블록을 생성하여 상기 블록을 블록체인에 추가하는 단계를 포함한다.

발명의 효과

[0020] 상술한 본 발명의 일 측면에 따르면, 블록체인 기반의 SDN 참여자 관리 시스템 및 방법을 제공함으로써, SDN에서 새로운 규칙을 배포하거나 네트워크를 구성하는 노드가 새롭게 네트워크에 참여하는 과정에서 유효성 검사 과정을 보장하여 악의적인 노드의 네트워크 가입을 제한할 수 있다.

[0021] 또한 본 발명의 블록체인 기반의 SDN 참여자 관리 시스템 및 방법을 제공함으로써, SDN 네트워크에 소속된 노드들의 지속적인 네트워크 참여 여부를 결정하여 SDN 네트워크 연결 정보를 안전하게 관리하는 것은 물론, 전체 시스템의 신뢰성을 보장하여 안정적으로 유지할 수 있다.

도면의 간단한 설명

[0022] 도 1은 본 발명의 일 실시예에 따른 블록체인 기반의 SDN 참여자 관리 시스템의 구성을 도시한 도면,

도 2는 본 발명의 일 실시예에 따른 블록체인 기반의 SDN 참여자 관리 시스템에서의 블록체인의 구조를 설명하기 위한 도면,

도 3은 본 발명의 일 실시예에 따른 블록체인 기반의 SDN 참여자 관리 방법을 설명하기 위한 흐름도,

도 4는 본 발명의 일 실시예에 따른 블록체인 기반의 SDN 참여자 관리 방법을 구체적으로 설명하기 위한 흐름도, 그리고

도 5 내지 도 8은 본 발명의 일 실시예에 따른 블록체인 기반의 SDN 참여자 관리 시스템의 효과를 설명하기 위한 실험 결과를 도시한 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0023] 후술하는 본 발명에 대한 상세한 설명은, 본 발명이 실시될 수 있는 특정 실시예를 예시로서 도시하는 첨부 도면을 참조한다. 이들 실시예는 당업자가 본 발명을 실시할 수 있기에 충분하도록 상세히 설명된다. 본 발명의 다양한 실시예는 서로 다르지만 상호 배타적일 필요는 없음이 이해되어야 한다. 예를 들어, 여기에 기재되어 있는 특정 형상, 구조 및 특성은 일 실시예와 관련하여 본 발명의 정신 및 범위를 벗어나지 않으면서 다른 실시예로 구현될 수 있다. 또한, 각각의 개시된 실시예 내의 개별 구성요소의 위치 또는 배치는 본 발명의 정신 및 범위를 벗어나지 않으면서 변경될 수 있음이 이해되어야 한다. 따라서, 후술하는 상세한 설명은 한정적인 의미로서 취하려는 것이 아니며, 본 발명의 범위는, 적절하게 설명된다면, 그 청구항들이 주장하는 것과 균등한 모든 범위와 더불어 첨부된 청구항에 의해서만 한정된다. 도면에서 유사한 참조부호는 여러 측면에 걸쳐서 동일하게

나 유사한 기능을 지칭한다.

- [0024] 본 발명에 따른 구성요소들은 물리적인 구분이 아니라 기능적인 구분에 의해서 정의되는 구성요소들으로써 각각이 수행하는 기능들에 의해서 정의될 수 있다. 각각의 구성요소들은 하드웨어 또는 각각의 기능을 수행하는 프로그램 코드 및 프로세싱 유닛으로 구현될 수 있을 것이며, 두 개 이상의 구성요소의 기능이 하나의 구성요소에 포함되어 구현될 수도 있을 것이다. 따라서 이하의 실시예에서 구성요소에 부여되는 명칭은 각각의 구성요소를 물리적으로 구분하기 위한 것이 아니라 각각의 구성요소가 수행되는 대표적인 기능을 암시하기 위해서 부여된 것이며, 구성요소의 명칭에 의해서 본 발명의 기술적 사상이 한정되지 않는 것임에 유의하여야 한다.
- [0025] 이하에서는 도면들을 참조하여 본 발명의 바람직한 실시예들을 보다 상세하게 설명하기로 한다.
- [0026] 도 1은 본 발명의 일 실시예에 따른 블록체인 기반의 SDN 참여자 관리 시스템(10)의 구성을 도시한 도면이다.
- [0027] SDN(Software-Defined Networking)은 차량 네트워크, 사물인터넷(Internet of Things, IOT) 네트워크, 에너지 관리 네트워크 등에 효과적으로 활용되고 있는 기술이다.
- [0028] 따라서 SDN을 구성하는 노드의 참여를 검증하고, SDN 연결의 신뢰성을 보장하여 전체적인 시스템을 안정적으로 유지할 필요가 있다.
- [0029] 종래의 SDN의 경우에는 SDN 컨트롤러에 의해 각 노드가 소프트웨어 애플리케이션 수준에서 수정 및 제어되어 관리되고 있다. 하지만 SDN 컨트롤러가 모든 노드의 신규 참여 및 기존 참여에 대한 정보를 확인할 수 없기 때문에 SDN에 참여하는 노드를 완전히 신뢰하기에는 어렵다는 문제가 있다.
- [0030] 본 실시예에 따른 블록체인 기반의 SDN 참여자 관리 시스템(10, 이하 시스템)은 이러한 SDN에서 새로운 규칙을 배포하거나 악의적인 노드의 네트워크 가입을 제한하고, 네트워크에 소속된 노드들의 지속적인 네트워크 참여 여부를 결정함으로써 네트워크의 신뢰성 및 안정성을 향상시키기 위해 마련된다.
- [0031] 따라서 본 실시예에 따른 시스템(10)은 복수의 노드(N)를 포함하여 구성되는 SDN에서 블록체인에 기반하여 노드(N)의 참여를 관리한다.
- [0032] 이러한 본 실시예에 따른 시스템(10)은 복수의 노드(N) 및 SDN 컨트롤러(300)를 포함할 수 있다. 그리고 시스템(10)은 블록체인 기반의 SDN 참여자 관리 방법을 수행하기 위한 소프트웨어(어플리케이션)가(이) 설치되어 실행될 수 있으며, 복수의 노드(N) 및 SDN 컨트롤러(300)는 블록체인 기반의 SDN 참여자 관리 방법을 수행하기 위한 소프트웨어(어플리케이션)에 의해 제어될 수 있다.
- [0033] 먼저 복수의 노드(N)는 SDN 컨트롤러(300)의 제어에 따라 패킷을 처리하는 기능을 수행하며, 이러한 복수의 노드(N)의 일 예로는 이동 통신 기지국, 기지국 제어기, 게이트웨이 장비, 유선 네트워크 스위치, 라우터, IoT 기기 등 다양한 형태로 구비될 수 있다.
- [0034] 그리고 본 실시예에 따른 복수의 노드(N)는 제1 노드(100) 및 제2 노드(200)를 포함할 수 있다.
- [0035] 먼저 제1 노드(100)는 복수의 노드(N) 중에서 SDN에 참여할 때 SDN 컨트롤러(300)로부터 부여받은 참여 시간을 블록이 생성되는 주기 미만의 시간으로 부여받은 노드일 수 있다.
- [0036] 또한 제1 노드(100)는 허용된 SDN의 참여 시간의 마감이 임박한 노드일 수 있다.
- [0037] 즉, 제1 노드(100)는 SDN 컨트롤러(300)에게 SDN의 참여 시간을 갱신 요청하는 요청 노드를 의미할 수 있다.
- [0038] 이러한 제1 노드(100)는 SDN에 참여 가능하도록 허용된 참여 시간을 갱신하기 위한 갱신 트랜잭션(Transaction)을 생성할 수 있다.
- [0039] 상기 제1 노드(100)에서 생성하는 갱신 트랜잭션은, 트랜잭션의 인덱스 현재 허용된 참여 시간, 제1 노드(100)와 현재 연결된 인접 노드의 수, 갱신할 참여 시간, 제1 노드(100)와 연결될 인접 노드의 수 중 적어도 하나를 포함할 수 있다.
- [0040] 그리고 제1 노드(100)는 생성된 갱신 트랜잭션의 검증을 위해 전체 네트워크에 생성된 갱신 트랜잭션을 브로드캐스트(Broadcast)하여 SDN의 참여를 요청할 수 있다.
- [0041] 제1 노드(100)가 갱신 트랜잭션을 생성하여 전파함으로써 참여 시간을 요청하고, 제1 노드(100)가 요청한 참여 시간을 부여받는 구체적인 과정에 대해서는 후술할 도 2에서 자세히 설명하기로 한다.
- [0042] 한편 제2 노드(200)는 브로드캐스트된 갱신 트랜잭션에 대한 검증을 수행하기 위해 마련될 수 있다.

- [0043] 이러한 제2 노드(200)는 복수의 노드(N) 중에서 SDN에 참여할 때 SDN 컨트롤러(300)로부터 부여받은 참여 시간을 블록이 생성되는 주기 이상의 시간으로 부여받은 노드일 수 있다.
- [0044] 또한 제2 노드(200)는 허용된 SDN의 참여 시간의 마감이 임박하지 않은 노드일 수 있다.
- [0045] 즉, 제2 노드(200)는 네트워크에 브로드캐스트된 갱신 트랜잭션에 대한 검증을 수행하는 검증 노드를 의미할 수 있다.
- [0046] 이러한 제2 노드(200)는 제1 노드(100)의 갱신 트랜잭션에 대해 검증 및 해싱(Hashing)을 수행할 수 있다.
- [0047] 제2 노드(200)는 갱신 트랜잭션을 사전에 마련되는 참여 시간 합의 알고리즘에 기초하여 검증하고, 검증이 완료된 갱신 트랜잭션을 댐풀(Mempool)로 전송할 수 있다.
- [0048] 구체적으로 제2 노드(200)는 참여 시간 합의 알고리즘에 따라 현재 타임스탬프(Time stamp)가 갱신 트랜잭션에 포함된 현재 허용된 참여 시간과 갱신할 참여 시간 사이인지를 확인하여 갱신 트랜잭션을 검증할 수 있다.
- [0049] 또한 제2 노드(200)는, 현재 연결된 인접 노드의 수와 연결될 인접 노드의 수에 기초한 제1 노드(100)의 네트워크 연결 상태를 확인하여 갱신 트랜잭션을 검증할 수 있다.
- [0050] 따라서 제2 노드(200)는, 상기 갱신 트랜잭션을 검증한 결과에 대응하는 비율이 사전에 설정된 임계 비율을 초과하면 해당 갱신 트랜잭션이 유효한 것으로 판단할 수 있다.
- [0051] 그리고 제2 노드(200)는 유효한 갱신 트랜잭션을 댐풀로 전송할 수 있다.
- [0052] 한편 SDN 컨트롤러(300)는 제1 노드(100) 및 제2 노드(200)를 포함하는 복수의 노드(N)와 연결되어 복수의 노드(N)를 관리 및 제어하기 위해 마련된다.
- [0053] 이러한 SDN 컨트롤러(300)는 제2 노드(200)가 갱신 트랜잭션을 검증한 결과에 따라 제1 노드(100)에게 갱신된 참여 시간을 부여할 수 있다.
- [0054] 또한 SDN 컨트롤러(300)는 SDN을 최초로 구성할 때 복수의 노드(N) 각각에게 참여 시간 및 참여 요청 가능 시간을 부여할 수 있다.
- [0055] 이때 SDN 컨트롤러(300)가 복수의 노드(N) 각각에게 부여하는 참여 시간 및 참여 요청 가능 시간의 합은 모두 일정할 수 있다.
- [0056] 그리고 SDN 컨트롤러(300)는 갱신 트랜잭션을 포함하는 블록을 생성하고, 생성된 블록을 블록체인에 추가할 수 있다.
- [0057] 구체적으로 SDN 컨트롤러(300)는 일정 주기마다 댐풀에 저장된 검증이 완료된 갱신 트랜잭션을 포함하는 블록을 생성하고, 생성된 블록을 블록체인에 추가할 수 있다.
- [0059] 이하에서는 도 2를 참고하여 본 실시예에 따른 시스템(10)에서 갱신 트랜잭션을 전파하여 참여 시간을 요청하고, 갱신 트랜잭션의 검증을 수행하며, 검증 결과로 갱신된 참여 시간을 부여하고, 블록을 생성하는 구체적인 과정에 대해서 설명하기로 한다.
- [0060] 도 2는 본 발명의 일 실시예에 따른 시스템(10)에서의 블록체인의 구조를 설명하기 위한 도면이다.
- [0061] 먼저 하기의 표 1에서와 같이 같이 SDN을 구성하기 위해 SDN 컨트롤러(300)는, 제1 노드(100) 및 제2 노드(200)를 포함하는 N개의 노드와 연결된다.

[0062] [표 1]

Algorithm 1 Node participation extension request process

1. For a node is taken occupied and reserved time by SDN controller according to genesis block B_0 :

$$T_i = T = T_{occupied,i} + T_{reserved,i}$$

2. For the 1-st timeslot:

if $T_{occupied,i} \leq T_{period}$ then

S_i request time extension as Algorithm 2.

else

S_i validate and hashing transaction from Algorithm2.

$$T_{occupied,requester}(1) = T_{occupied,requester} + \alpha$$

$$T_{reserved,requester}(1) = T_{reserved,requester} - \alpha$$

Find nonce for valid hash of new block B_1 .

end if

3. For the i -th timeslot:

if $T_{occupied,i,i} \leq T_{period}$ then

S_i request time extension as Algorithm 2.

else

S_i validate and hashing transaction from Algorithm2.

$$T_{occupied,requester}(i) = T_{occupied,requester}(i-1) + \alpha$$

$$T_{reserved,requester}(i) = T_{reserved,requester}(i-1) - \alpha$$

Find nonce for valid hash of new block B_1 .

end if

[0063]

[0064] 그리고 제1 노드(100) 및 제2 노드(200)를 포함하는 N개의 모든 노드는 초기 설정으로 참여 시간 $T_{occupied,i}$ 와 참여 요청 가능 시간 $T_{reserved,i}$ 을 SDN 컨트롤러(300)로부터 부여받을 수 있다.

[0065] 제1 노드(100) 및 제2 노드(200)의 최초 점유 시간은 제네시스 블록 B_0 생성에 따라 임의로 설정되나, 각 노드(N)가 부여받은 총 참여 시간과 참여 요청 가능 시간의 합 T 는 모두 일정할 수 있다. 이는 아래의 수학적 식 1로 표현될 수 있다.

[0066] [수학적 식 1]

[0067]
$$T_i = T = T_{occupied,i} + T_{reserved,i}$$

[0068]
$$T_{total} = n \times T = \sum_{i=n} (T_{occupied,i} + T_{reserved,i})$$

[0069] 이때 만약, 상술한 바와 같이 제네시스 블록 B_0 이 생성된 이후 두 번째 블록의 생성까지의 타임스탬프 T_{period} 미만의 참여 시간을 부여받은 노드는 하기의 표 2에서와 같이 제1 노드(100), 즉 요청 노드($S_{request}$)로써 참여 시간의 갱신을 위한 갱신 트랜잭션 TX_i 를 생성할 수 있다.

[0070] [표 2]

Algorithm 2 SDN participation consensus

```

1. Node  $S_{request}$  generates a request transaction  $TX_i$  to extend the network participation time:
 $TX_i = [t_{pp}, n_c, t_{np}, n_n]$ 
2. The node  $S_{request}$  broadcasts the generated transaction  $TX_i$  to other  $N - 1$  nodes in the network.
3. Other nodes validate the transaction  $TX_i$ :
  if  $t_{pp} \leq t$  and  $t_{np} \geq t$  then
    if  $n_c \leq N - 1$  and  $n_n \leq N - 1$  then
      validated = True
    else
      validated = False
    end if
  else
    validated = False
  end if
4. Each node transmits the validation result  $V_{S_i}$  of the transaction to the mempool  $M$ .
5. Validation results are stored in mempool  $M$  until the block generation cycle.
6. The controller confirms the validity of the transactions that occurred up to timestamp  $t$ :
  for  $i < t$  do
    if  $r_{vi} > R_v$  then
       $b_{new} \leftarrow b_{new} + TX_i$ 
    End if
  End for
7. Create the corresponding transactions as a block and connect with the previous block:
 $B_i \leftarrow B_i + b_{new}$ 

```

[0071]

[0072] 이러한 갱신 트랜잭션 TX_i 는 트랜잭션의 인덱스 i , 현재 허용된 참여 시간 t_{pp} , 제1 노드(100)와 현재 연결된 인접 노드의 수 n_c , 갱신할 참여 시간 t_{np} , 제1 노드(100)와 연결될 인접 노드의 수 n_n 중 적어도 하나를 포함할 수 있다.

[0073] 그리고 생성된 갱신 트랜잭션의 유효성이 표 2에서와 같이 참여 시간 합의 알고리즘을 통해 검증되면 해당 제1 노드(100)는 갱신된 참여 시간을 부여받을 수 있다.

[0074] 그리고 검증이 완료된 해당 갱신 트랜잭션은 SDN 컨트롤러(300)가 생성하는 블록에 포함되며, 생성된 블록은 제네시스 블록 이후에 연결될 블록에 포함시킬 수 있다. 이러한 과정을 반복하여 SDN 컨트롤러(300)는 이전에 생성된 블록에 다음 블록을 연결하여 블록체인의 형태로 네트워크 시간 요청 트랜잭션들에 대한 정보를 저장할 수 있다.

[0075] 반면에 상술한 바와 같이 제네시스 블록 B_0 이 생성된 이후 두 번째 블록의 생성까지의 타임스탬프 T_{period} 이상의 참여 시간을 부여받은 노드는 표 2에서와 같이 제2 노드(200), 즉 검증 노드로써 제1 노드(100)의 갱신 트랜잭션에 대해 검증 및 해싱을 수행할 수 있다.

[0076] 구체적으로 갱신 트랜잭션 TX_i 를 수신한 제2 노드(200)는 현재 허용된 참여 시간 t_{pp} 가 현재 타임스탬프 t 이전이고, 갱신할 참여 시간 t_{np} 이 현재 타임스탬프 t 이후 인지 확인한다.

[0077] 그리고나서 제2 노드(200)는 해당 제1 노드와 현재 연결된 인접 노드의 수 n_c 와 제1 노드(100)와 연결될 인접 노드의 수 n_n 를 사용하여 해당 제1 노드(100)가 네트워크 연결 상태에 대한 정보를 가지고 있는지를 확인할 수 있다.

[0078] 보다 구체적으로 상기 표 2에서와 같이 제2 노드(200)는 현재 연결된 인접 노드의 수 n_c 와 연결될 인접 노드의 수 n_n 가 전체 네트워크의 노드의 수 이하인지를 확인한다.

[0079] 이러한 과정을 통해 제2 노드(200)는 갱신 트랜잭션이 현재 네트워크의 타임스탬프에 대한 정보와 연결 정보를

정확하게 반영하였는지를 검증할 수 있다.

- [0080] 그리고 제2 노드(200)는 해당 갱신 트랜잭션이 유효한 것으로 검증된 검증 결과의 비율 r_{vid1} 미리 정해진 파라미터 검증 비율 R_v 보다 크면 해당 갱신 트랜잭션을 유효한 트랜잭션으로 확인하고, 검증 결과를 맴폴 b_{new} 로 전송할 수 있다.
- [0081] 한편 SDN 컨트롤러(300)는 맴폴에 저장된 유효한 트랜잭션에 대한 검증 결과를 바탕으로 해당 제1 노드(100)에 SDN에 참여할 수 있는 갱신된 참여 시간을 부여할 수 있다.
- [0082] 그리고 SDN 컨트롤러(300)는 검증된 트랜잭션을 종합한 맴폴에 담긴 트랜잭션들을 이용하여 SDN에 참여하는 노드(N)의 네트워크 참여와 관련된 블록을 생성하고, 생성된 블록을 연결하여 체인을 형성할 수 있다.
- [0083] 이러한 SDN 컨트롤러에서 형성되는 블록체인의 구조는 도 2에 도시된 바와 같다. 도 2에 도시된 바와 같이 블록의 구조는 크게 블록 헤더(Header)와 바디(Body)로 구성될 수 있다.
- [0084] 블록 헤더에는 블록의 고유 식별자인 블록번호, 이전 블록의 해시값, 타임스탬프, 난이도(논스(Nonce) 값) 등의 정보가 포함될 수 있다.
- [0085] 그리고 바디에는 블록에 저장되는 실제 트랜잭션 데이터가 담기는 거래 데이터가 포함될 수 있다.
- [0086] 블록 내의 모든 트랜잭션 데이터는 수많은 검증이 완료된 트랜잭션을 머클 트리(Merkle Tree)라는 자료 구조로 구성된 후, 연결된 트랜잭션의 해시 값을 반복적으로 계산하여 최종 루트 값을 저장한다. 여기서 머클 루트 값은 블록 헤더에 저장되어 블록 헤더를 구성하는 요소로써 사용된다.
- [0087] 또한, 블록 생성의 난이도를 의미하는 논스 값은 블록체인에서 일정한 난이도를 충족하는 블록의 해시값을 찾아내는 작업 증명(Proof of Work) 알고리즘을 이용한 블록의 생성에 사용될 수 있다.
- [0088] 그리고 블록 생성에 반복적으로 사용되는 해시함수(SHA-256)는 블록체인 구조의 무결성을 검증하고 전체 보안을 유지할 수 있다.
- [0089] 이러한 블록을 복수개 포함하는 블록체인은 각각의 블록이 서로 연결되는 체인형태로 마련되기 때문에 이전 블록의 해시값이 다음 블록의 헤더에 포함되어 블록 간 연결성과 무결성을 보장할 수 있다.
- [0090] 이에 따라 제2 노드(200)는 결과적으로 머클 트리 루트(Merkle Tree Root)를 도출하여 이를 기반으로 이전 블록의 정보들을 포함한 정보를 통해 논스(Nonce)를 해제하여 블록 생성에 기여하는 노드로 동작할 수 있다.
- [0091] 따라서 본 실시예에 따른 시스템(10)은 SDN에서 새로운 규칙을 배포하거나 네트워크를 구성하는 노드가 새롭게 네트워크에 참여하는 과정에서 유효성 검사 과정을 보장하여 악의적인 노드의 네트워크 가입을 제한할 수 있다.
- [0092] 또한 SDN 네트워크에 소속된 노드들의 지속적인 네트워크 참여 여부를 결정하여 SDN 네트워크 연결 정보를 안전하게 관리하는 것은 물론, 전체 시스템의 신뢰성을 보장하여 안정적으로 유지할 수 있다.
- [0094] 한편 도 3 및 도 4는 본 발명의 일 실시예에 따른 블록체인 기반의 SDN 참여자 관리 방법을 설명하기 위한 흐름도로써, 본 발명의 일 실시예에 따른 블록체인 기반의 SDN 참여자 관리 방법은 도 1에 도시된 시스템(10)과 실질적으로 동일한 구성 상에서 진행되므로, 도 1의 시스템(10)과 동일한 구성요소에 대해 동일한 도면 부호를 부여하고, 반복되는 설명은 생략하기로 한다.
- [0095] 본 실시예에 따른 블록체인 기반의 SDN 참여자 관리 방법은, SDN의 참여를 요청하는 단계(S110), 검증을 수행하는 단계(S130) 및 참여 시간을 부여하고 블록을 생성하여 블록을 블록체인에 추가하는 단계(S150)를 포함할 수 있다.
- [0096] SDN의 참여를 요청하는 단계(S110)에서는 복수의 노드 중 하나인 제1 노드(100)가, SDN에 참여 가능하도록 허용된 참여 시간을 갱신하기 위한 갱신 트랜잭션을 생성할 수 있다.
- [0097] 그리고 SDN의 참여를 요청하는 단계(S110)에서는 제1 노드(100)가 생성된 갱신 트랜잭션을 네트워크 상에 전파하여 SDN의 참여를 요청할 수 있다.
- [0098] 한편 검증을 수행하는 단계(S130)에서는 복수의 노드 중 하나인 제2 노드(200)가 수신된 갱신 트랜잭션에 대한 검증을 수행할 수 있다.

- [0099] 그리고 참여 시간을 부여하고 블록을 생성하여 블록을 블록체인에 추가하는 단계(S150)에서는 SDN 컨트롤러(300)가, 검증 결과에 따라 제1 노드(100)에게 갱신된 참여 시간을 부여할 수 있다.
- [0100] 또한 참여 시간을 부여하고 블록을 생성하여 블록을 블록체인에 추가하는 단계(S150)에서는 SDN 컨트롤러(300)가 검증이 완료된 갱신 트랜잭션을 포함하는 블록을 생성할 수 있다.
- [0101] 그리고 참여 시간을 부여하고 블록을 생성하여 블록을 블록체인에 추가하는 단계(S150)에서는 SDN 컨트롤러(300)가 생성된 블록을 블록체인에 추가할 수 있다.
- [0102] 이하에서는 이러한 본 실시예에 따른 블록체인에 기반한 SDN 참여자 관리 시스템이 수행하는 SDN 참여자 관리 방법을 수행하는 과정을 도 4를 참조하여 보다 구체적으로 설명하기로 한다.
- [0103] 먼저 SDN 내부의 모든 노드들은 SDN 컨트롤러(300)의 제네시스 블록 정보에 따라 네트워크 참여 시간(예컨대 점유시간)과 요청 가능 시간(예약 시간)을 할당 받을 수 있다(S205).
- [0104] 그리고 나서 SDN을 구성하는 노드 중에 네트워크 참여 시간이 블록을 생성하는 시간보다 짧으면(S210-예) 해당 노드는 제1 노드(100)로써 네트워크 추가 참여를 위한 갱신 트랜잭션을 생성할 수 있다(S220).
- [0105] 그리고 제1 노드(100)는 갱신 트랜잭션을 자신을 제외한 SDN을 구성하는 모든 노드에게 브로드캐스트할 수 있다(S225).
- [0106] 반면 SDN을 구성하는 노드 중에 네트워크 참여 시간이 블록을 생성하는 시간보다 길면(S210-아니오), 해당 노드는 네트워크에 참여하는 검증 노드, 즉 제2 노드(200)로써 생성된 트랜잭션에 대한 검증을 수행하고, 블록 생성을 위한 해싱 문제를 해결할 수 있다(S215).
- [0107] 이를 위해 제2 노드(200)는 생성된 트랜잭션에 포함되는 현재 허용된 참여 시간과 갱신할 참여 시간을 현재 타임스탬프와 비교하여 검증할 수 있다(S230).
- [0108] 즉 도 6에 도시된 바와 같이 제2 노드(200)는 현재 허용된 참여 시간이 현재 타임스탬프 이전이고, 갱신할 참여 시간이 현재 타임스탬프 이후인지를 확인한다(S235).
- [0109] 만약 현재 허용된 참여 시간이 현재 타임스탬프 이전이고, 갱신할 참여 시간이 현재 타임스탬프 이후인 조건을 만족하면(S235-예), 제2 노드(200)는 노드 수에 대한 유효성 검사를 수행할 수 있다(S240).
- [0110] 즉, 도 4에 도시된 바와 같이 제2 노드(200)는 제1 노드(100)와 현재 연결된 인접 노드의 수와 제1 노드(100)와 연결될 인접 노드의 수가 전체 네트워크 노드 수 이하인지를 확인한다(S245).
- [0111] 제1 노드(100)와 현재 연결된 인접 노드의 수와 제1 노드(100)와 연결될 인접 노드의 수가 전체 네트워크 노드 수 이하이면(S245-예), 제2 노드(200)는 해당 갱신 트랜잭션은 검증이 완료된 트랜잭션으로써, 해당 갱신 트랜잭션에 대한 검증 결과를 맴폴에 저장할 수 있다(S250).
- [0112] 이후 SDN 컨트롤러(300)는 맴폴 내부에 저장된 갱신 트랜잭션들의 유효성을 확인할 수 있다(S255).
- [0113] 그리고 SDN 컨트롤러(300)는 해당 제1 노드(100)의 네트워크 참여 시간을 갱신 요청한 시간만큼 추가로 부여할 수 있다(S260).
- [0114] 그리고 나서 SDN 컨트롤러(300)는 블록 생성 주기에 맞춰 맴폴의 네트워크 참여 시간 검증 결과에 대한 갱신 트랜잭션들을 포함하는 블록을 생성할 수 있다(S265).
- [0115] 만약 현재 허용된 참여 시간이 현재 타임스탬프 이전이고, 갱신할 참여 시간이 현재 타임스탬프 이후인 조건을 만족하지 않거나(S235-아니오), 제1 노드(100)와 현재 연결된 인접 노드의 수와 제1 노드(100)와 연결될 인접 노드의 수가 전체 네트워크 노드 수를 초과하면(S245-아니오), 제2 노드(200)는 해당 갱신 트랜잭션의 검증 실패에 따라 해당 갱신 트랜잭션을 미반영한다(S270). 따라서 해당 갱신 트랜잭션을 생성한 제1 노드(100)는 네트워크 참여 시간이 제한될 수 있다(S270).
- [0117] 한편 도 5 내지 도 8은 본 발명의 일 실시예에 따른 블록체인 기반의 SDN 참여자 관리 시스템의 효과를 설명하기 위한 실험 결과를 도시한 그래프이다.
- [0118] 도 5 및 도 6은 다수의 노드와 SDN 컨트롤러로 구성된 시뮬레이션 환경에서 합의 알고리즘의 합의 도달 시간에

대한 검증을 진행한 결과를 나타낸 그래프이다.

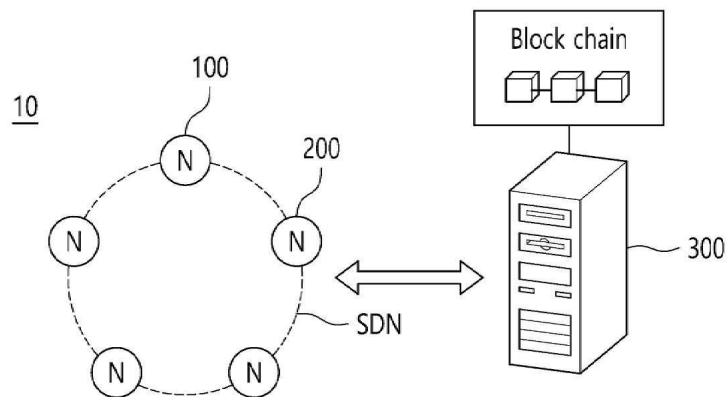
- [0119] 구체적으로 도 5에서는 200개의 노드로 구성된 네트워크에서 선행된 0(leading zeros)의 개수로 표현되는 난이도 변경에 따라 검증 결과를 얻는데 필요한 시간을 도출한 결과이다.
- [0120] 도 5에 도시된 바와 같이 가장 낮은 난이도인 선행 0이 3개인 경우 전체 노드의 약 50%의 유효한 검증 결과를 얻는데 약 1.2초, 합의율 90%는 약 1.9초가 소요되는 것을 확인하였다. 한편 선행 0이 4개인 경우는 50%와 90%의 유효한 검증 결과를 얻는데 약 2.9초와 5.2초가 소요되었다.
- [0121] 마지막으로 선행 0이 5개인 가장 어려운 난이도의 경우 50%의 유효한 검증 결과를 얻는데 약 26.6초, 90%의 유효한 검증 결과를 얻는데 약 66.3초가 소요되었다.
- [0122] 도 6은 블록 생성을 위한 난이도를 고정하고 시뮬레이션 환경에서 사용된 노드의 수를 조절하며 유효한 검증 결과를 얻는데 걸리는 시간을 나타내는 그래프이다.
- [0123] 도 6에 도시된 바와 같이 사용된 노드의 수가 증가할수록 선형적으로 합의 도달에 소요되는 시간이 증가하였다.
- [0124] 한편 도 7 및 도 8은 시스템(10)에서 관리하는 전체 시간이 분산된 노드들 사이에 고르게 분포되어 있는지, 프로세스가 지속적으로 유지될 수 있는지를 검증한 결과이다.
- [0125] 도 7에서 실선은 모든 노드의 참여 시간($T_{occupied,i}$)의 합을 의미하고, 점선은 참여 요청 가능 시간 $T_{reserved,i}$ 의 합을 의미하는 것으로 각각 66147.3초와 23852.7초이다. 도 7에 도시된 바와 같이 전체 네트워크에 속하는 노드들의 참여 시간 및 참여 요청 가능 시간이 안정적으로 유지되는 것을 확인할 수 있다.
- [0126] 한편 도 8은 잔여 네트워크 참여 시간이 가장 임박한 노드의 시간 값을 타임 슬롯별로 측정하여 표시한 도면이다.
- [0127] 도 8에 도시된 바와 같이 가장 낮은 값으로 동작하는 노드에서도 최소 40초의 시간이 보장되므로 본 실시예에 따른 시스템(10)의 모든 노드가 이탈 없이 참여할 수 있다.
- [0128] 이와 같은 본 발명의 블록체인 기반의 SDN 참여자 관리 방법은 다양한 컴퓨터 구성요소를 통하여 수행될 수 있는 프로그램 명령어의 형태로 구현되어 컴퓨터 판독 가능한 기록 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능한 기록 매체는 프로그램 명령어, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다.
- [0129] 상기 컴퓨터 판독 가능한 기록 매체에 기록되는 프로그램 명령어는 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 분야의 당업자에게 공지되어 사용 가능한 것일 수도 있다.
- [0130] 컴퓨터 판독 가능한 기록 매체의 예에는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM, DVD 와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 ROM, RAM, 플래시 메모리 등과 같은 프로그램 명령어를 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다.
- [0131] 프로그램 명령어의 예에는, 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드도 포함된다. 상기 하드웨어 장치는 본 발명에 따른 처리를 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.
- [0132] 이상에서는 본 발명의 다양한 실시예에 대하여 도시하고 설명하였지만, 본 발명은 상술한 특정의 실시예에 한정되지 아니하며, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 다양한 변형실시가 가능한 것은 물론이고, 이러한 변형실시들은 본 발명의 기술적 사상이나 전망으로부터 개별적으로 이해되어져서는 안될 것이다.

부호의 설명

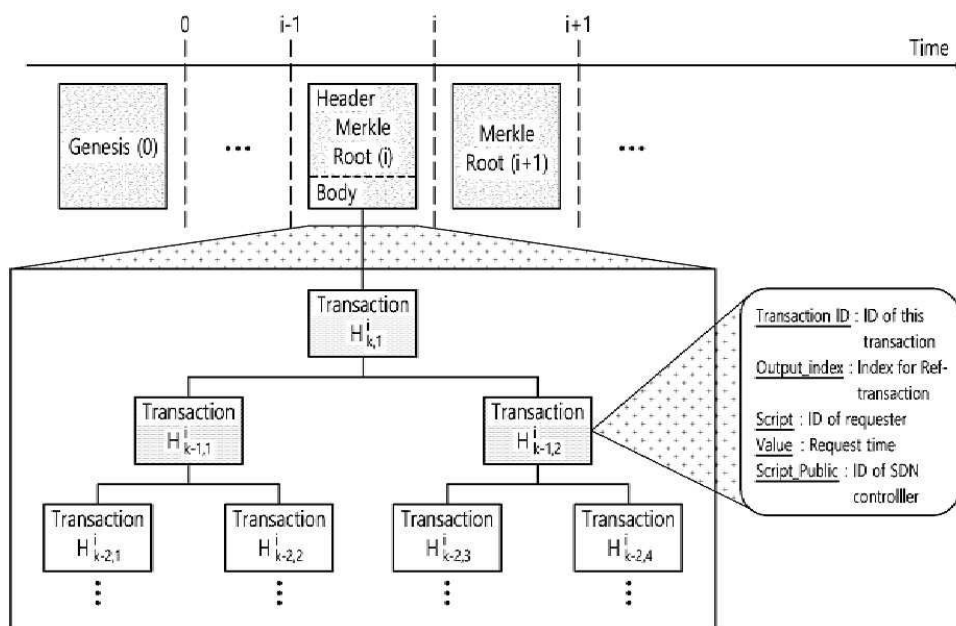
- [0133] 10 : 블록체인 기반의 SDN 참여자 관리 시스템
N : 노드 100 : 제1 노드
200 : 제2 노드 300 : SDN 컨트롤러

도면

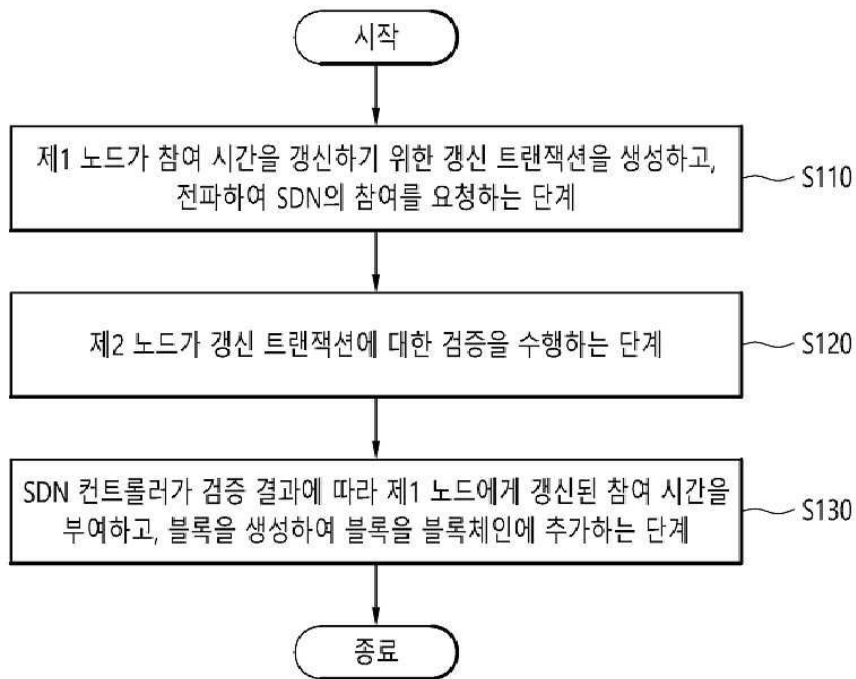
도면1



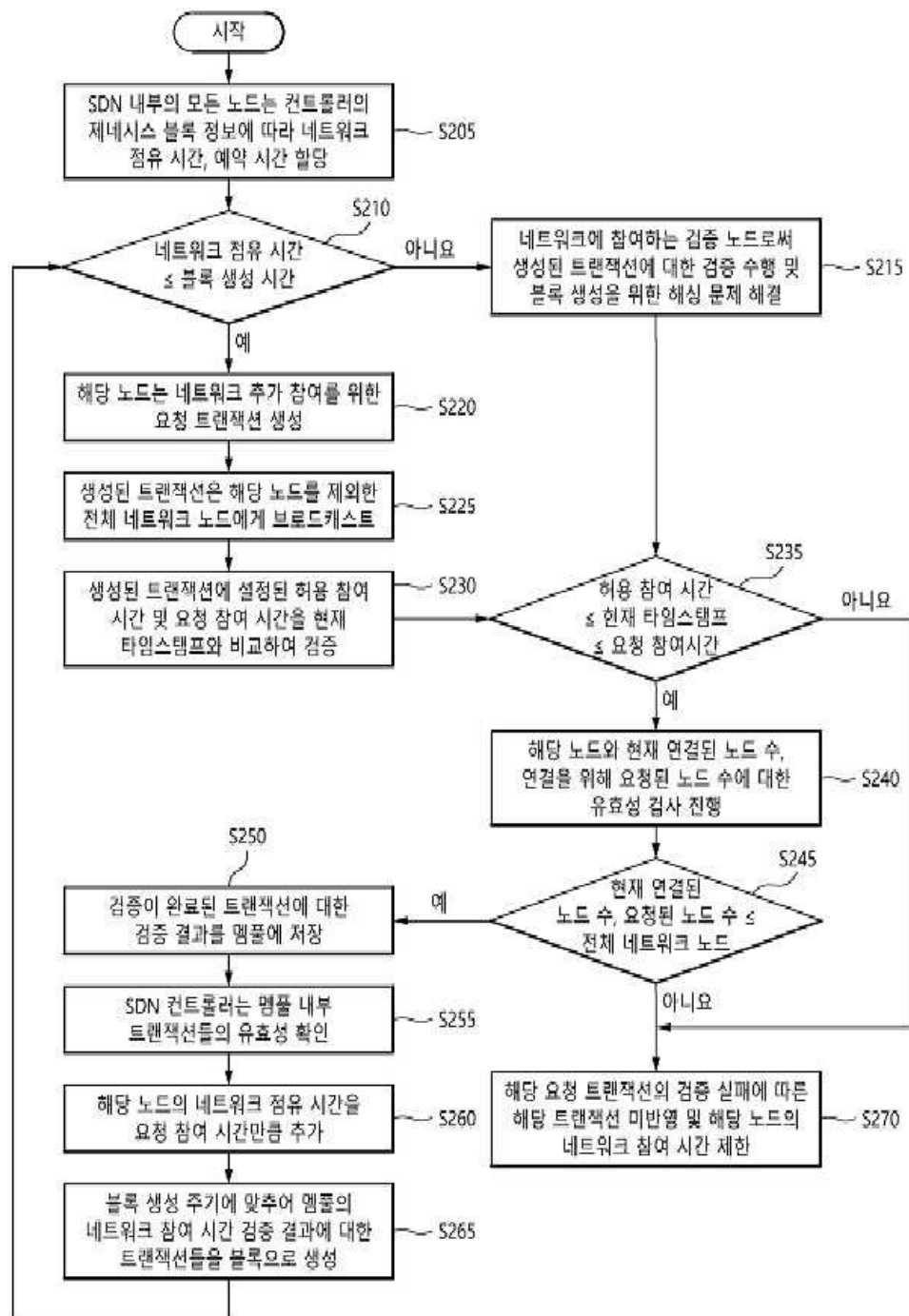
도면2



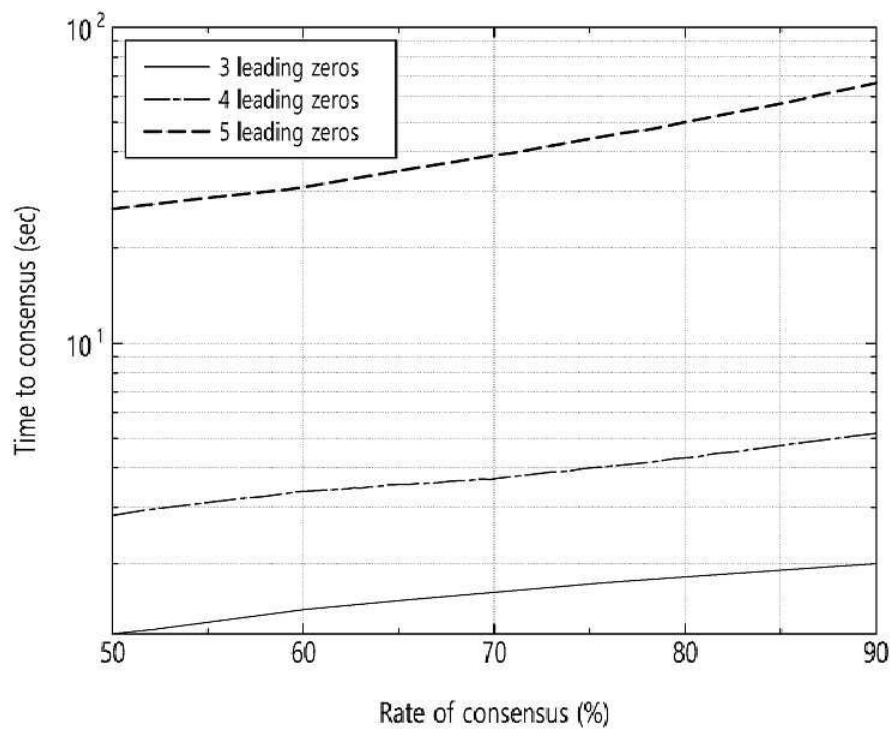
도면3



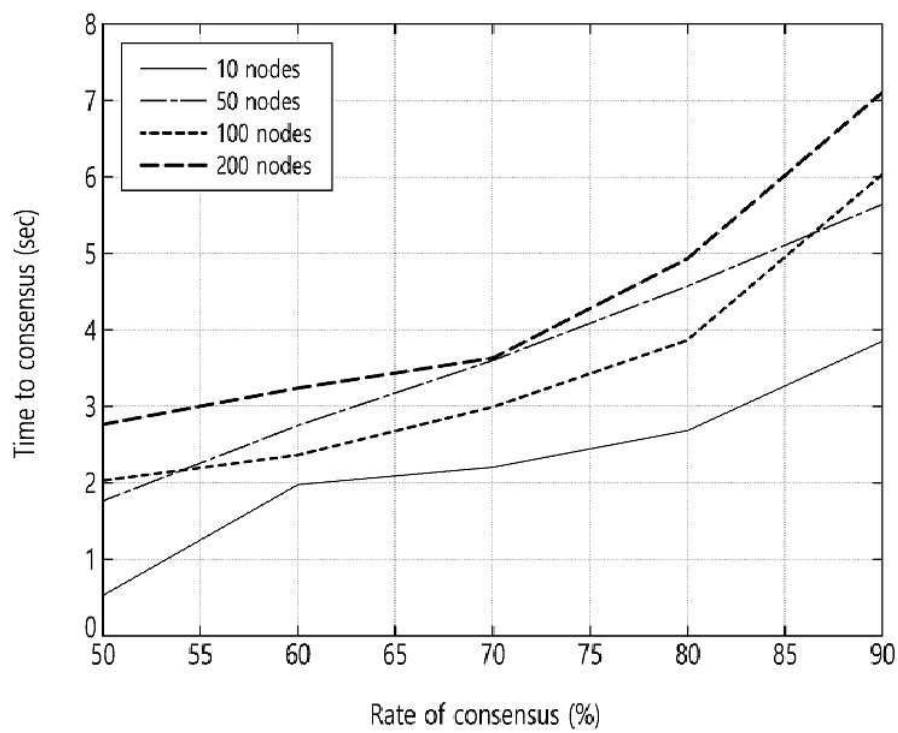
도면4



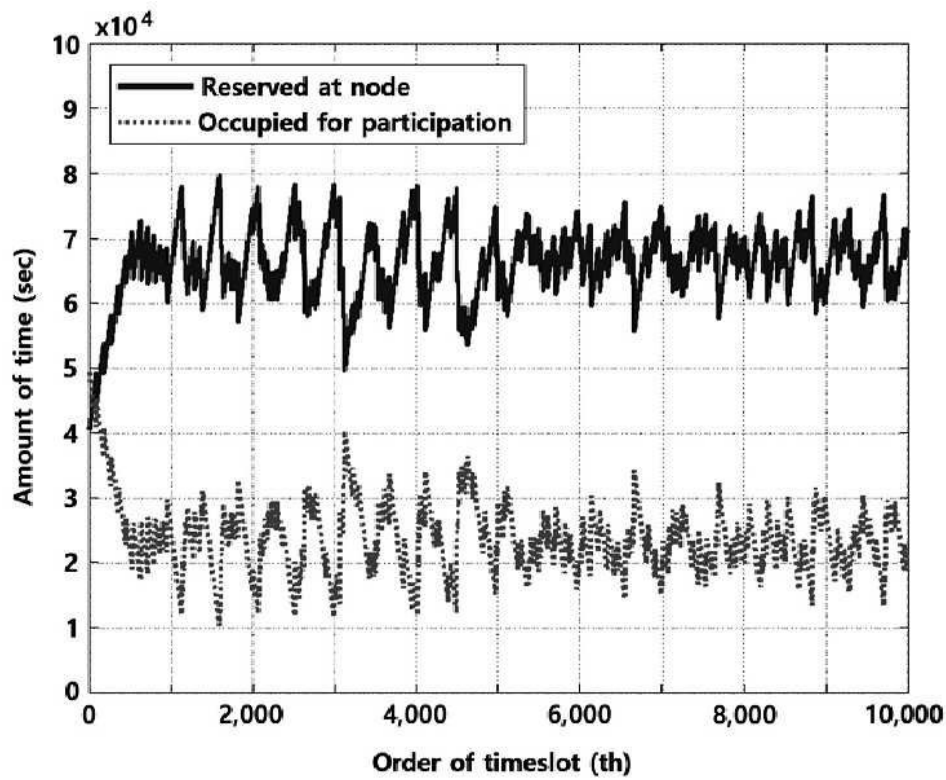
도면5



도면6



도면7



도면8

