(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0022993
(43) 공개일자 2024년02월20일

(51) 국제특허분류(Int. Cl.)
H04L 67/1087 (2022.01) *H04L 67/1042* (2022.01)
H04L 67/1074 (2022.01) *H04L 67/1097* (2022.01)
H04L 9/00 (2022.01)
(52) CPC특허분류
H04L 67/1093 (2022.05)
H04L 67/1044 (2022.05)
(21) 출원번호 10-2023-0103642
(22) 출원일자 2023년08월08일
심사청구일자 2023년08월08일
(30) 우선권주장
1020220101450 2022년08월12일 대한민국(KR)

(71) 출원인
포항공과대학교 산학협력단
경상북도 포항시 남구 청암로 77 (지곡동)
(72) 발명자
박찬익
경상북도 포항시 남구 청암로 77
정우창
경상북도 포항시 남구 청암로 77
(74) 대리인
특허법인이상

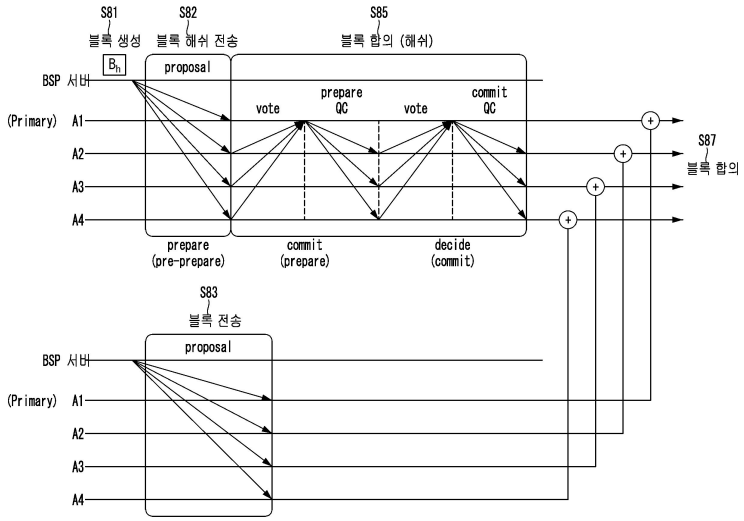
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **블록체인 합의 단계에서 탈중앙화를 강화시키기 위한블록 처리 방법 및 장치**

(57) 요 약

블록체인 합의 과정에서의 탈중앙화 강화와 블록체인 플랫폼의 성능 확장성을 높일 수 있는 블록 처리 방법 및 장치가 개시된다. 블록 처리 방법은 블록체인 네트워크에서 블록을 생성하는 블록 생성 노드로부터 생성된 블록의 해시값을 수신하는 단계, 및 상기 블록의 해시값을 이용하여 블록 합의를 시작하는 단계를 포함한다.

대 표 도



(52) CPC특허분류
H04L 67/1074 (2022.05)
H04L 67/1097 (2022.05)
H04L 9/50 (2022.05)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711152571
과제번호	2021-0-00484-002
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	데이터 경제를 위한 블록체인 기술개발(R&D)
연구과제명	노드 간 메시지 전달과 합의를 위한 최적 경로 네트워크프로토콜 기술개발
기 여 율	1/1
과제수행기관명	포항공과대학교 산학협력단
연구기간	2022.01.01 ~ 2022.12.31

명 세 서

청구범위

청구항 1

블록체인 네트워크의 합의 참여 노드들 중 어느 하나에 의해 수행되는 블록체인 합의 과정에서의 탈중앙화를 강화하는 블록 처리 방법으로서,

블록체인 네트워크에서 블록을 생성하는 블록 생성 노드로부터 생성된 블록의 해시값을 수신하는 단계; 및

상기 블록의 해시값을 이용하여 블록 합의를 시작하는 단계;

를 포함하는 블록 처리 방법.

청구항 2

청구항 1에 있어서,

상기 해시값을 수신하는 단계는 상기 해시값을 포함하는 상기 블록의 블록 헤더 정보를 수신하는 것을 포함하는 블록 처리 방법.

청구항 3

청구항 1에 있어서,

상기 블록 생성 노드로부터 전송되는 상기 블록을 수신하는 단계를 더 포함하는 블록 처리 방법.

청구항 4

청구항 3에 있어서,

상기 해시값을 이용한 블록 합의에 사용된 해시값과 상기 블록을 수신하는 단계를 통해 획득한 블록의 해시값을 비교하여 상기 블록 또는 상기 블록 합의의 유효성을 검증하는 단계를 더 포함하는 블록 처리 방법.

청구항 5

청구항 3에 있어서,

상기 해시값을 수신하는 단계를 통해 상기 해시값이 획득되면 타이머를 일정한 시간 또는 제1 시간으로 설정하는 단계를 더 포함하는 블록 처리 방법.

청구항 6

청구항 5에 있어서,

상기 타이머가 상기 제1 시간에서 타임아웃되면, 현재의 블록 생성 노드가 악의적인 블록 생성 노드인 것으로 판단하는 단계를 더 포함하는 블록 처리 방법.

청구항 7

청구항 6에 있어서,

상기 현재의 블록 생성 노드가 악의적인 블록 생성 노드일 때, 상기 현재의 블록 생성 노드를 다른 블록 생성 노드로 교체하는 단계를 더 포함하는 블록 처리 방법.

청구항 8

블록체인 네트워크에 포함된 감사 네트워크의 감사자들 중 어느 하나에 의해 수행되는 블록체인 합의 과정에서의 탈중앙화를 강화하는 블록 처리 방법으로서,

블록체인 네트워크의 블록생성자들에 속한 액티브 블록생성자로부터 상기 액티브 블록생성자에 의해 생성된 블

록의 해시값을 수신하는 단계; 및

상기 블록의 해시값을 이용하여 블록 합의를 시작하는 단계;

를 포함하는 블록 처리 방법.

청구항 9

청구항 8에 있어서,

상기 해시값을 수신하는 단계는 상기 해시값을 포함하는 상기 블록의 블록 헤더 정보를 수신하는 것을 포함하는 블록 처리 방법.

청구항 10

청구항 8에 있어서,

상기 액티브 블록생성자로부터 전송되는 상기 블록을 수신하는 단계를 더 포함하는 블록 처리 방법.

청구항 11

청구항 10에 있어서,

상기 해시값을 이용한 블록 합의에 사용된 해시값과 상기 블록을 수신하는 단계를 통해 획득한 블록의 해시값을 비교하여 상기 블록 또는 상기 블록 합의의 유효성을 검증하는 단계를 더 포함하는 블록 처리 방법.

청구항 12

청구항 10에 있어서,

상기 해시값을 수신하는 단계를 통해 상기 해시값이 획득되면 타이머를 일정한 시간 또는 제1 시간으로 설정하는 단계를 더 포함하는 블록 처리 방법.

청구항 13

청구항 12에 있어서,

상기 타이머가 상기 제1 시간에서 타임아웃되면, 상기 액티브 블록생성자가 악의적인 블록생성자인 것으로 판단하는 단계; 및

상기 액티브 블록생성자가 악의적인 블록생성자일 때, 상기 액티브 블록생성 자를 스탠바이 블록생성자들 중 하나로 교체하는 단계;

를 더 포함하는 블록 처리 방법.

청구항 14

블록체인 네트워크에 포함된 감사 네트워크의 감사자들 중 어느 하나로서 블록체인 합의 과정에서의 탈중앙화를 강화하는 블록 처리 장치에 있어서,

적어도 하나의 명령을 저장하는 메모리에 연결되어 상기 적어도 하나의 명령을 실행하는 프로세서를 포함하고,

상기 적어도 하나의 명령에 의해 상기 프로세서가,

블록체인 네트워크의 블록생성자들에 속한 액티브 블록생성자로부터 상기 액티브 블록생성자에 의해 생성된 블록의 일부 데이터를 수신하는 단계; 및

상기 블록의 일부 데이터를 이용하여 블록 합의를 시작하는 단계;

를 수행하는 포함하는 블록 처리 장치.

청구항 15

청구항 14에 있어서,

상기 블록의 일부 데이터는 상기 블록의 해시값 또는 상기 해시값을 포함하는 블록 헤더 정보를 포함하는 블록

처리 장치.

청구항 16

청구항 15에 있어서,

상기 블록 헤더 정보는 블록체인 플랫폼들마다 서로 다른 블록 처리 장치.

청구항 17

청구항 15에 있어서,

상기 프로세서가, 상기 액티브 블록생성자로부터 전송되는 상기 블록을 수신하는 단계를 더 수행하는 블록 처리 장치.

청구항 18

청구항 17에 있어서,

상기 프로세서가, 상기 해시값을 이용한 블록 합의에 사용된 해시값과 상기 블록을 수신하는 단계를 통해 획득한 블록의 해시값을 비교하여 상기 블록 또는 상기 블록 합의의 유효성을 검증하는 단계를 더 수행하는 블록 처리 장치.

청구항 19

청구항 15에 있어서,

상기 프로세서가, 상기 해시값을 수신하는 단계를 통해 상기 해시값이 획득되면 타이머를 일정한 시간 또는 제1 시간으로 설정하는 단계를 더 수행하는 블록 처리 장치.

청구항 20

청구항 19에 있어서,

상기 프로세서는, 상기 타이머가 상기 제1 시간에서 타임아웃될 때, 상기 액티브 블록생성자가 악의적인 블록생성자인 것으로 판단하고, 상기 액티브 블록생성 자를 스탠바이 블록생성자들 중 하나로 교체하는 단계를 더 수행하는 블록 처리 장치.

발명의 설명

기 술 분 야

[0001] 본 발명은 블록체인 플랫폼의 성능을 향상시키는 기술에 관한 것으로, 보다 상세하게는, 블록체인 합의 단계에서 탈중앙화를 강화시킴으로써 블록체인 플랫폼의 성능을 향상시키는 블록 처리 방법 및 장치에 관한 것이다.

배 경 기 술

[0002] 블록체인(Blockchain)은 P2P(Peer to Peer) 네트워크를 통해서 관리되는 분산 데이터베이스 혹은 분산 원장 기술의 한 형태로, 데이터를 중앙 서버 한 곳에 저장하는 것이 아니라 블록체인 네트워크에 연결된 여러 컴퓨터에 블록들을 체인 형태로 연결하여 저장 및 보관하는 데이터 구조나 이러한 데이터 구조에 적용되는 기술을 지칭한다.

[0003] 블록체인 네트워크에 참여하는 모든 노드들(Nodes)은 노드들 간의 합의 알고리즘을 바탕으로 동일한 데이터를 공유한다. 블록체인의 블록은 노드들이 공유하고 있는 데이터를 각각 포함한 트랜젝션들(Transactions)의 집합으로 구성되고, 이전 블록의 해시(Hash) 값을 담고, 다른 블록들과 체인 형식으로 연결된다.

[0004] 블록체인은 비허가형 블록체인(Permissionless Blockchain)과 허가형 블록체인(permissioned Blockchain)으로 구분된다. 비허가형 블록체인은 모든 참여자와 노드가 네트워크에 참여할 수 있고 읽기 및 쓰기, 합의 등 모든 권한을 가진다. 한편, 허가형 블록체인은 검증된 참여자와 노드만이 네트워크에 참여할 수 있고, 특정 참여자와 노드들에 한해 지정된 권한을 가진다.

[0005] 기존 블록체인 플랫폼에서 하나의 블록을 처리하는 데에는 블록 전송 단계, 합의 단계, 및 결과 전파 단계의 3

단계 과정이 필요하고, 이러한 3단계 과정은 순차적으로 진행될 필요가 있다. 순차적으로 진행되는 3단계 과정은 블록체인 플랫폼의 성능에 영향을 미치는 요인이 되기도 한다.

[0006] 블록체인 시스템을 활용하는 기존 응용 서비스에는 증권 거래(Stock Exchange), 공급망(Supply Chain), 온라인 결재(Online Payment), 건강관리(Health Care), 지능형 전력망(Smart Grid), 스마트시티(Smart City), 사물인터네서(Internet of Things, IoT) 등이 있으며, 이에 더하여 새로운 응용 서비스로는 대체불가한토큰(Non-Fungible Token, NFT), 탈중앙화 금융(Decentralized Finance, Defi, 디파이), 게임(Game), 메타버스(Metaverse) 등과 같은 서비스들이 등장하고 있다. 새로 등장하는 응용 서비스에 맞추어 블록체인 사용자 수가 급증하는 추세이고, 그에 따른 블록체인 응용 서비스의 성능 요구사항이 높아지고 있다.

[0007] 이와 같이, 블록체인 응용 서비스에 대한 성능 요구사항이 높아짐에 따라서 블록체인 시스템에 대한 고성능이 요구되고 있다.

발명의 내용

해결하려는 과제

[0008] 본 발명은 전술한 요구에 부응하기 위해 도출된 것으로, 본 발명의 목적은 블록체인 플랫폼에서 하나의 블록을 처리하는 데 필요한 블록 전송 단계와 합의 단계를 중첩시켜 블록 처리 시간을 단축시키고 합의 단계에서 탈중앙화를 강화시킬 수 있는 블록 처리 방법 및 장치를 제공하는데 있다.

[0009] 본 발명의 또 다른 목적은 블록 전송 단계와 합의 단계를 중첩시키기 위해 블록 일부 데이터를 기반으로 합의 단계를 시작할 수 있는 블록 처리 방법 및 장치를 제공하는데 있다.

[0010] 본 발명의 또 다른 목적은 블록체인 합의 단계를 블록 가용성(block availability) 문제로 정의하여 처리할 수 있는 블록 처리 방법 및 장치를 제공하는데 있다.

과제의 해결 수단

[0011] 상기 기술적 과제를 해결하기 위한 본 발명의 일 측면에 따른 블록 처리 방법은, 블록체인 블록의 처리 과정이 블록 전송, 합의, 결과 전송의 3단계로 구성되는 블록체인 플랫폼에서, 블록 전송 단계와 합의 단계를 중첩(overlapping)시키는 것을 포함하고, 이에 의해 블록체인 합의 단계에서 탈중앙화를 강화한다.

[0012] 상기 방법은, 중첩을 적용할 때 합의 단계 완결을 위해 블록 데이터 전체를 수신해야 하는데, 블록 데이터 수신 단계에서 미리 설정된 타임아웃값 내에 블록 수신을 성공하지 못할 경우 즉, 블록 가용성이 만족되지 않을 경우에, 해당 높이의 블록 생성을 재요청하는 단계를 진행하고 정상적인 블록 생성 과정을 통해 결국 블록을 성공적으로 수신하는 단계를 포함하도록 구성되거나, 혹은 해당 높이 블록에 대해서는 합의되지 않은 것으로 결정하고, 빈(Blank) 블록으로 블록체인을 생성하는 단계를 포함하도록 구성될 수 있다.

[0013] 상기 방법에서, 하나의 합의 단계는, 블록 해쉬값 등 블록 일부 데이터에 대한 합의와 전체 블록에 대한 수신을 병렬처리 가능한 과정들로 구분되고, 두 과정들이 모두 충족해야/완료되어야 해당 블록에 대한 합의 단계가 종료되도록 구성될 수 있다.

[0014] 상기 방법은, 블록 데이터를 수신하고, 해당 블록에 대한 합의 단계를 진행하지 않고, 블록 데이터에 대한 일부분의 값 예를 들어, 블록 해시값 혹은 블록 헤더를 해당 블록의 합의를 시작하기 위해 먼저 전송한 후, 해당 블록 전체를 전송하여 합의 단계를 처리하는 과정을 포함할 수 있다.

[0015] 상기 방법에서, 합의를 수행하는 노드는 블록 데이터의 수신과 합의 단계를 중첩하여 진행할 수 있다. 이것은 블록 데이터 크기보다 블록 해시값이나 블록 헤더 정보 크기가 훨씬 작기 때문에, 블록 합의 단계에 대한 가용 시간이 블록 전송 시간에 제한되지 않고 독립적으로 혹은 병렬적으로 진행될 수 있다. 결과적으로 블록체인 노드들 간 네트워크 대역폭 차이가 크더라도 블록 해쉬값처럼 크기가 작은 정보는 빨리 전파될 수 있으므로 합의 단계를 빠른 시점에 시작하고, 이와 동시에 블록 데이터 수신 단계를 중첩하여 진행함으로써, 블록체인 합의 처리 성능을 향상시킬 수 있다.

[0016] 상기 기술적 과제를 해결하기 위한 본 발명의 다른 측면에 따른 블록체인 블록 처리 장치는, 블록체인 합의 단계에서 탈중앙화를 강화하기 위한 장치로서, 작업 증명(proof of work, PoW)과 같이 블록 생성과 합의 과정이 통합되어 있는 경우는 적용하지 못하나, 비잔틴 장애 허용(Byzantine fault tolerance, BFT) 합의 혹은 권위 증명(proof of authority, PoA) 등 블록 생성과 합의 과정이 별도의 과정으로 정의되어 있는 모든 블록체인 플랫

폼에 적용될 수 있다.

[0017] 상기 장치는, 블록체인 네트워크에서 사용자 트랜잭션을 스마트컨트랙트 실행함으로써 블록 데이터를 생성하고, 블록체인 네트워크 상 합의 참여 노드들에 전파하는 기능을 담당하는 노드를 제1 엔티티로 포함하고, 제1 엔티티로부터 수신한 해시값과 블록 데이터에 기초하여 합의 단계를 진행하고 합의가 완료된 후 트랜잭션 처리 결과를 회신하는 기능을 담당하는 합의 참여 노드를 제2 엔티티로 포함할 수 있다.

[0018] 상기 장치는, 블록체인 플랫폼인 감사 체인(Audit chain) 상에 구현될 수 있다. 즉, 상기 장치는, 작업증명(PoW)을 채택하는 블록체인 플랫폼을 제외한 현재 모든 블록체인 플랫폼에 적용될 수 있다. 감사 체인에서는 제1 엔티티를 블록 생성자(block service provider, BSP)로 포함하고, 합의 참여 노드인 제2엔티티를 감사자(Auditor 혹은 A-node)로 포함할 수 있다. 그리고 제2 엔티티들 간 합의 알고리즘은 부분적으로 동기화(Partially Synchronous)하는 네트워크 모델을 가정한 비잔틴 장애 허용(Byzantine fault tolerance, BFT) 합의 알고리즘을 기반으로 할 수 있다.

[0019] 상기 제1 엔티티인 BSP는, 임의의 클라이언트로부터 트랜잭션을 수신하고, 해당 트랜잭션에 대한 유효성 검사를 수행한 후, 유효한 트랜잭션을 모아 블록을 생성할 수 있고, 생성한 블록을 제2 엔티티인 감사자에게 전파할 수 있다.

[0020] 상기 제2 엔티티는, 제1 엔티티로부터 블록 데이터를 수신하여 합의를 진행하고 클라이언트에게 합의 결과를 전송할 수 있다. 제2 엔티티인 감사자는 제1 엔티티인 BSP의 동작을 감시하고, BSP가 악의적인 공격을 한다고 판단한 경우, 현재의 BSP를 스탠바이 BSP 중에서 선택하여 새로운 정상적 BSP로 변경하는 BSP 교체 프로토콜을 진행할 수 있다.

[0021] 상기 방법은, 제1 엔티티가 블록을 생성하고 생성한 블록을 제2 엔티티로 전송할 때, 블록 전체를 전송하지 않고 블록의 일부 데이터, 즉 블록 해쉬값이나 이를 포함한 블록 헤더 정보를 먼저 제2 엔티티로 전송하는 단계를 포함할 수 있다. 이 경우, 제2 엔티티는 먼저 수신한 블록 일부 데이터에 기반하여 합의 과정을 시작하고, 합의 과정을 수행하면서 병렬적으로 해당 블록을 수신하는 단계를 포함할 수 있다. 결과적으로, 제2 엔티티는 블록 합의와 블록 수신의 두 개의 서로 다른 프로세스를 동시에 진행할 수 있다. 여기서, 블록 합의 프로세스는 제1 엔티티로부터 블록의 일부 데이터 예컨대 블록 해시를 수신하고 수신한 블록 해시를 토대로 합의 단계를 진행하는 것을 포함하고, 블록 수신 프로세스는 제1 엔티티로부터 블록의 전체 데이터를 수신하는 것을 포함한다.

[0022] 상기 기술적 과제를 해결하기 위한 본 발명의 또 다른 측면에 따른 블록 처리 방법은, 블록체인 네트워크의 합의 참여 노드들 중 어느 하나에 의해 수행되는 블록체인 합의 과정에서의 탈중앙화를 강화하는 블록 처리 방법으로서, 블록체인 네트워크에서 블록을 생성하는 블록 생성 노드로부터 생성된 블록의 해시값을 수신하는 단계; 및 상기 블록의 해시값을 이용하여 블록 합의를 시작하는 단계를 포함한다.

[0023] 상기 해시값을 수신하는 단계는 상기 해시값을 포함하는 상기 블록의 블록 헤더 정보를 수신하는 것을 포함할 수 있다.

[0024] 상기 블록 처리 방법은, 상기 블록 생성 노드로부터 전송되는 상기 블록을 수신하는 단계를 더 포함할 수 있다.

[0025] 상기 블록 처리 방법은, 상기 해시값을 이용한 블록 합의에 사용된 해시값과 상기 블록을 수신하는 단계를 통해 획득한 블록의 해시값을 비교하여 상기 블록 또는 상기 블록 합의의 유효성을 검증하는 단계를 더 포함할 수 있다.

[0026] 상기 블록 처리 방법은, 상기 해시값을 수신하는 단계를 통해 상기 해시값이 획득되면 타이머를 일정한 시간 또는 제1 시간으로 설정하는 단계를 더 포함할 수 있다.

[0027] 상기 블록 처리 방법은, 상기 타이머가 상기 제1 시간에서 타임아웃되면, 현재의 블록 생성 노드가 악의적인 블록 생성 노드인 것으로 판단하는 단계를 더 포함할 수 있다.

[0028] 상기 블록 처리 방법은, 상기 현재의 블록 생성 노드가 악의적인 블록 생성 노드일 때, 상기 현재의 블록 생성 노드를 다른 블록 생성 노드로 교체하는 단계를 더 포함할 수 있다.

[0029] 상기 기술적 과제를 해결하기 위한 본 발명의 또 다른 측면에 따른 블록체인 블록 처리 방법은, 블록체인 네트워크에 포함된 감사 네트워크의 감사자들 중 어느 하나에 의해 수행되는 블록체인 합의 과정에서의 탈중앙화를 강화하는 블록 처리 방법으로서, 블록체인 네트워크의 블록생성자들에 속한 액티브 블록생성자로부터 상기 액티브 블록생성자에 의해 생성된 블록의 해시값을 수신하는 단계; 및 상기 블록의 해시값을 이용하여 블록 합의를

시작하는 단계를 포함한다.

[0030] 상기 해시값을 수신하는 단계는 상기 해시값을 포함하는 상기 블록의 블록 헤더 정보를 수신하는 것을 포함할 수 있다.

[0031] 상기 블록 처리 방법은, 상기 액티브 블록생성자로부터 전송되는 상기 블록을 수신하는 단계를 더 포함할 수 있다.

[0032] 상기 블록 처리 방법은, 상기 해시값을 이용한 블록 합의에 사용된 해시값과 상기 블록을 수신하는 단계를 통해 획득한 블록의 해시값을 비교하여 상기 블록 또는 상기 블록 합의의 유효성을 검증하는 단계를 더 포함할 수 있다.

[0033] 상기 블록 처리 방법은, 상기 해시값을 수신하는 단계를 통해 상기 해시값이 획득되면 타이머를 일정한 시간 또는 제1 시간으로 설정하는 단계를 더 포함할 수 있다.

[0034] 상기 블록 처리 방법은, 상기 타이머가 상기 제1 시간에서 타임아웃되면, 상기 액티브 블록생성자가 악의적인 블록생성자인 것으로 판단하는 단계; 및 상기 액티브 블록생성자가 악의적인 블록생성자일 때, 상기 액티브 블록생성 자를 스탠바이 블록생성자들 중 하나로 교체하는 단계를 더 포함할 수 있다.

[0035] 상기 기술적 과제를 해결하기 위한 본 발명의 또 다른 측면에 따른 블록체인 블록 처리 장치는, 블록체인 네트워크에 포함된 감사 네트워크의 감사자들 중 어느 하나로서 블록체인 합의 과정에서의 탈중앙화를 강화하는 블록 처리 장치에 있어서, 적어도 하나의 명령을 저장하는 메모리에 연결되어 상기 적어도 하나의 명령을 실행하는 프로세서를 포함한다. 그리고 상기 적어도 하나의 명령에 의해 상기 프로세서가, 블록체인 네트워크의 블록생성자들에 속한 액티브 블록생성자로부터 상기 액티브 블록생성자에 의해 생성된 블록의 일부 데이터를 수신하는 단계; 및 상기 블록의 일부 데이터를 이용하여 블록 합의를 시작하는 단계를 수행하도록 구성된다.

[0036] 상기 블록의 일부 데이터는 상기 블록의 해시값 또는 상기 해시값을 포함하는 블록 헤더 정보를 포함할 수 있다.

[0037] 상기 블록 헤더 정보는 블록체인 플랫폼들마다 서로 다를 수 있다.

[0038] 상기 프로세서는, 상기 액티브 블록생성자로부터 전송되는 상기 블록을 수신하는 단계를 더 수행할 수 있다.

[0039] 상기 프로세서는, 상기 해시값을 이용한 블록 합의에 사용된 해시값과 상기 블록을 수신하는 단계를 통해 획득한 블록의 해시값을 비교하여 상기 블록 또는 상기 블록 합의의 유효성을 검증하는 단계를 더 수행할 수 있다.

[0040] 상기 프로세서는, 상기 해시값을 수신하는 단계를 통해 상기 해시값이 획득되면 타이머를 일정한 시간 또는 제1 시간으로 설정하는 단계를 더 수행할 수 있다.

[0041] 상기 프로세서는, 상기 타이머가 상기 제1 시간에서 타임아웃될 때, 상기 액티브 블록생성자가 악의적인 블록생성자인 것으로 판단하고, 상기 액티브 블록생성 자를 스탠바이 블록생성자들 중 하나로 교체하는 단계를 더 수행할 수 있다.

발명의 효과

[0042] 전술한 본 발명의 실시예들에 의하면, 블록 전송 단계와 합의 단계를 중첩하여 수행함으로써 블록체인 합의 단계에서 탈중앙화를 강화할 수 있다. 즉, 제1 엔티티의 블록체인 블록의 전송 시, 제2 엔티티가 블록 데이터 전체를 수신하지 않고 블록 해쉬값 등 블록과 관련된 일부 정보를 수신하여 합의 단계를 시작하도록 함으로써, 블록 데이터 전체를 수신한 후에 합의 단계를 시작하는 기존 블록 처리 방식에 비해 블록 처리율을 크게 높일 수 있다.

[0043] 또한, 본 발명에 의하면, 합의 참여 노드들인 제2 엔티티들 간 네트워크 대역폭 차이가 크더라도, 블록에 비해 상대적으로 크기가 매우 작은 블록 해쉬값을 수신하는 시간은 제2 엔티티들 간에 크게 차이가 나지 않고, 또한 블록 전체 데이터를 수신하는 시간은 합의 단계를 진행하는 시간과 중첩될 수 있으므로, 결과적으로 블록 처리 시간을 줄여 블록체인 성능 확장성을 높일 수 있다.

[0044] 또한, 본 발명에 의하면, 블록 처리율을 보장하는 블록체인 플랫폼 예컨대, 클레이튼(KLAYTN)과 같이 1초당 1개의 블록 처리 시간을 보장하는 퍼블릭 블록체인 플랫폼에서, 블록 전송 단계와 합의 단계를 병렬적으로 처리할 수 있도록 함으로써, 실질적으로 합의 단계의 처리 허용 시간을 증가시킬 수 있고, 이에 따라 합의 참여 노드 개수를 증가시키거나, 합의 참여 노드들간 물리적 위치를 자유롭게 선정하거나 확대할 수 있어 탈중앙화 요소를

강화할 수 있다.

[0045] 또한, 본 발명에 의하면, 허가형 블록체인 플랫폼에서 블록 전송 단계와 합의 단계의 중첩에 필요한 블록 가용성을 일정 시간 범위 내에서 블록 생성자 교체를 통해 보장함으로써, 실질적으로 합의 단계의 처리 허용 시간을 증가시킬 수 있고, 이에 따라 합의 참여 노드 개수를 증가시키거나, 합의 참여 노드들간 물리적 위치를 자유롭게 선정하거나 확대할 수 있어 탈중앙화 요소를 강화할 수 있다.

도면의 간단한 설명

[0046] 도 1은 본 발명의 일 실시예에 따른 블록 처리 방법을 채용할 수 있는 블록체인 시스템을 설명하기 위한 개략적인 구조도이다.

도 2는 도 1의 블록체인 시스템에서 생성되고 합의를 위해 전송되는 블록의 구성을 설명하기 위한 블록도이다.

도 3은 도 2의 블록에 대한 일반적인 3단계의 처리 과정을 설명하기 위한 도면이다.

도 4는 도 1의 블록체인 시스템에 적용할 수 있는 본 실시예의 블록 처리 방법의 주요 원리를 설명하기 위한 도면이다.

도 5는 본 발명의 다른 실시예에 따른 블록 처리 방법을 채용할 수 있는 블록체인 시스템에 대한 개념도이다.

도 6은 도 5의 블록체인 시스템에 채용할 수 있는 블록 생성자(block service provider, BSP)의 구성을 예시한 블록도이다.

도 7은 도 5의 블록체인 시스템에 적용가능한 비교예의 블록 처리 과정을 설명하기 위한 도면이다.

도 8은 도 5의 블록체인 시스템에 적용할 수 있는 본 실시예의 블록 처리 과정을 설명하기 위한 도면이다.

도 9는 도 5의 블록체인 시스템에 적용할 수 있는 본 실시예의 비잔틴 BSP 대응 블록 처리 과정을 설명하기 위한 도면이다.

도 10은 본 발명의 또 다른 실시예에 따른 블록 처리 장치에 대한 개략적인 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0047] 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.

[0048] 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.

[0049] 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.

[0050] 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.

[0051] 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.

[0052] 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.

[0053] 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.

[0054] 도 1은 본 발명의 일 실시예에 따른 블록 처리 방법을 채용할 수 있는 블록체인 시스템을 설명하기 위한 개략적인 구조도이다. 도 2는 도 1의 블록체인 시스템에서 생성되고 합의를 위해 전송되는 블록의 구성을 설명하기 위한 블록도이다. 도 3은 도 2의 블록에 대한 일반적인 3단계의 처리 과정을 설명하기 위한 도면이다. 그리고 도 4는 도 1의 블록체인 시스템에 적용할 수 있는 본 실시예의 블록 처리 방법의 주요 원리를 설명하기 위한 도면이다.

[0055] 도 1을 참조하면, 블록체인 시스템은 클라이언트(100), 블록 생성 노드(200a) 및 합의 참여 노드들(300a)을 포함한다. 클라이언트(100)는 클라이언트 게이트웨이(Client Gateway)로 지칭될 수 있다. 블록 생성 노드(200a)와 합의 참여 노드들(300a)을 포함하는 블록체인 네트워크(500)의 노드들은 피투피(P2P: peer to peer) 오버레이 네트워크(overlay network)를 통해 서로 연결될 수 있다. 블록 생성 노드(200a)는 제1 엔티티로, 합의 참여 노드들 각각은 제2 엔티티로 각각 지칭될 수 있다.

[0056] 클라이언트(100)는 사용자 단말(미도시)로부터 사용자 트랜젝션을 받고 받은 사용자 트랜젝션을 블록 생성 노드(200a)로 전달한다(S20). 사용자 트랜젝션은 클라이언트 트랜젝션(Client TXs)으로 지칭될 수 있다. 클라이언트는 클라이언트 SDK(Software Development Kit) 및 취합기(Aggregator)를 구비할 수 있다.

[0057] 클라이언트 SDK는 사용자 단말로부터 호출 인터페이스를 통해 수신한 스마트 컨트랙트 실행 인자들을 트랜젝션 제출 형태로 변환하고 이를 블록체인 네트워크(500)의 블록 생성 노드(200a)에 제출할 수 있다.

[0058] 취합기는 제출한 트랜젝션에 대한 응답을 블록체인 네트워크(500)로부터 수신할 수 있다. 여기서, 트랜젝션 응답(TX Reply)은 블록 생성 노드(200a)가 트랜젝션을 실행한 직후에 전송하는 이벤트, 그리고 합의 참여 노드들(300a)가 해당 트랜젝션에 대해 검증 직후에 전송하는 이벤트를 포함할 수 있다. 취합기는 이벤트 취합기 또는 취합기 모듈로 지칭될 수 있다.

[0059] 블록 생성 노드(200a)는 블록체인 네트워크(500)에서 블록 생성과 블록 전송을 담당할 수 있다. 블록 생성 노드(200a)는 클라이언트 SDK로부터 수신한 클라이언트 트랜젝션(Client TXs)을, 상태 데이터베이스(database, DB) 등에 저장되어 있는 스마트 컨트랙트(250)를 통해 실행하고, 그 결과를 포함한 블록(292)을 만들고 이를 블록체인(290)에 기록할 수 있다. 블록 생성 노드(200a)는 사용자 단말에서 직접 암호학적으로 서명된 트랜젝션을 블록(292)에 저장할 때 임의로 변경할 수 없다.

[0060] 합의 참여 노드들(300a)은 블록 생성 노드(200a)에서 P2P 오버레이 네트워크를 통해 받은 블록에 대하여 합의를 통해 유효성을 검증하고 합의 결과를 전파할 수 있다.

[0061] 한편, 블록 생성 노드(200a)가 합의 참여 노드들(300a)로 전송하는 블록(292)은 도 2에 나타낸 바와 같이 블록 헤더(2921) 및 블록 바디를 구비할 수 있다. 블록 헤더(2921)는 타임스탬프(timestamp), 해시값(hash value), 트랜젝션 의존성 정보를 포함할 수 있다. 그리고, 블록 바디는 복수의 트랜젝션들 예를 들어 제1 트랜젝션(트랜젝션 #1), 제2 트랜젝션(트랜젝션 #2) 및 제n 트랜젝션(트랜잭션 #n)을 포함할 수 있다.

[0062] 이 경우, 블록체인 네트워크(500)는 일반적인 3단계의 블록 처리 과정들을 수행할 수 있다. 즉, 도 3에 나타낸 바와 같이 블록체인 네트워크(500)는 블록체인을 위한 블록 처리를 위해 해당 블록의 모든 데이터를 각 합의 참여 노드로 전송하는 블록 전송 단계(S31), 블록 전송 단계(S31)를 통해 해당 블록의 모든 데이터의 수신을 각 합의 참여 노드에서 완료한 후에 합의 참여 노드들에 의해 수행되는 블록 합의 단계(S33) 및 합의 참여 노드들에 의해 수행된 블록 합의 결과를 블록체인 네트워크 내 노드들에게 전파하는 결과 전파 단계(S35)를 순차적으로 일정 시간 또는 제1 시간(T1) 동안에 수행하여 해당 블록을 처리할 수 있다.

[0063] 이러한 비교예의 블록 처리 과정에서 합의 노드들 간의 지정학적 위치에서의 거리 관계나 네트워크 상태 등에 따른 전송 지연 등의 환경에 따라 블록 처리 시간은 크게 영향을 받을 수 있고, 이것은 블록체인 시스템의 성능에 좋지 않은 영향을 미칠 수 있다.

[0064] 이에 본 실시예의 블록 처리 방법은 도 4에 나타낸 바와 같이 블록 전송 단계(S41)과 블록 합의 단계(S43)를 중첩(overlapping)시켜 블록 처리 시간(T2)을 일반적인 블록 처리 시간(도 3의 T1 참조)보다 현저히 단축시켜 블록 처리율을 높이고 블록체인 시스템의 블록 합의에서의 탈중앙화 성능을 향상시킨다.

[0065] 이를 위해, 본 실시예의 블록 처리 방법은 블록 생성 노드가 블록 생성 과정에서 얻은 블록 해쉬 또는 블록 해쉬를 포함하는 블록 헤더 정보를 제1 시점(t0)에 P2P 오버레이 네트워크를 통해 먼저 합의 참여 노드들로 전송하고(S40), 그에 의해 각 합의 참여 노드에서 해당 블록의 데이터 전부를 수신하기 전 제2 시점(t1)에 합의 과정을 우선적으로 시작하도록 하며, 그리고 전술한 블록 해쉬나 블록 헤더 정보의 전송 후에, 블록 생성 노드가, 생성된 블록을 합의 참여 노드들로 전송하도록 구성될 수 있다(S41).

[0066] 전술한 경우, 합의 참여 노드들은 블록 전송 단계(S41)를 통해 블록의 모든 데이터를 받기 전인 제1 시점(t0)에 블록 해쉬에 기초하여 블록 합의 단계(S43)을 빠르게 시작할 수 있고, 블록 합의를 처리하는 도중인 제2 시점(t1)에 해당 블록의 모든 데이터를 받고, 받은 블록 내 해시값 등을 통해 해당 블록의 유효성을 검증한 후, 블록 내 데이터에 기초하여 처리 중인 블록 합의를 완료할 수 있다. 블록 합의가 완료되면, 블록 합의 결과는 각 합의 참여 노드로부터 블록체인 네트워크 내 다른 노드들로 각각 전파될 수 있다(S45).

[0067] 이러한 구성에 의하면, 본 실시예의 블록 처리 방법은 블록체인 네트워크 내 합의 참여 노드들이 블록 합의를 처리하는데 있어서 각 합의 참여 노드의 지정학적 위치에 상대적으로 민감하지 않고 상대적으로 확장가능하게 확보된 블록 합의의 처리 시간 동안에 기존 대비 빠르게 블록 합의를 처리할 수 있다.

[0069] 부언하면, 블록체인 플랫폼에서 블록체인 성능은 블록 처리율 즉, 초당 블록 합의 갯수로 표현될 수 있다. 블록 처리율에 영향을 미치는, 하나의 블록을 처리하는 과정은 블록 전송 단계, 합의 단계, 및 결과 전파 단계의 3단계들로 구성되고 순차적 진행이 필요하다. 이에 따라 하나의 블록을 생성하는 것에 따라 블록체인 성능이 결정될 수 있다. 특히, 블록 합의 단계에서 탈중앙화를 높이기 위해서는, 합의 참여 노드들의 지리적 위치가 글로벌하게 배치되거나, 합의 참여 노드의 개수를 증가시킬 수 있다.

[0070] 이에 본 발명에서는 블록체인 플랫폼이 블록 처리율을 보장하기 위해 블록 합의 단계의 탈중앙화 요소를 제한하는 상황을 해결한다. 즉, 순차적으로 진행되는 3단계의 블록 처리 과정 중 블록 생성 및 전송 단계와 합의 단계를 중첩 (overlapping)시킴으로써, 좀더 구체적으로는 블록 전송 단계와 합의 단계를 중첩시킴으로써, 합의 단계에 허용되는 시간을 확대하고, 이를 통해 합의 단계의 탈중앙화를 높일 수 있다. 즉, 본 실시예에 의하면, 블록 처리율을 보장하면서 합의 참여 노드들의 물리적 위치를 글로벌하게 배치하거나 합의 참여 노드 개수를 증가시킬 수 있다.

[0071] 특히 최근 블록체인 기술에서는 데이터를 신뢰성 있게 저장하는 것을 핵심적인 기술로 부각되고 있다. 블록체인이란 거래 내역과 같은 데이터를 네트워크에 참여하는 사용자들이 분산하여 저장하고 처리하는 기술을 의미한다. 블록체인 기술을 실제 적용하고 있는 플랫폼들로는, 예시로 들자면, 비트코인(Bitcoin), 이더리움(Ethereum), 이오스(EOS), 텐더민트(Tendermint), 헤데라(Hedera), 하이퍼레저 패브릭(Hyperledger Fabric), 아이오타(IOTA) 등이 존재한다.

[0072] 블록체인은 모든 네트워크 참여자들이 서로 다른 데이터를 공유하며 검증 가능하다는 장점이 있는데, 데이터를 신뢰성 있게 공유해야 하므로 블록체인의 네트워크 참여자 간의 합의 알고리즘을 통해 이러한 신뢰성을 보장한다. 그러나 이러한 신뢰성을 보장하기 위한 합의 알고리즘은 모든 네트워크 참여자들이 참여해야 하고, 전체 블록 데이터를 공유해야 하므로 오버헤드가 매우 크고, 중앙화된 데이터 저장, 처리 기술보다 트랜잭션 처리 속도(transaction processing speed, TPS)가 현저하게 느리다는 단점이 있다.

[0073] 더욱이, 비트코인과 이더리움 같은 블록체인 플랫폼은 작업증명이라는 합의 알고리즘을 사용하여 블록체인의 신뢰성을 보장하는데, 이는 작업증명의 난이도에 따라 블록 생성 주기가 달라지며, 비트코인의 경우와 같이 길게는 약 10분, 이더리움의 경우와 같이 짧게는 약 12초의 주기를 갖게 된다. 그러나 이는 작업증명 문제를 푸는 채굴 과정을 거치게 되고 채굴 과정은 많은 양의 전기를 소모하게 된다. 즉 채굴과정을 통해 유지되는 블록체인 플랫폼은 해당 플랫폼을 유지하기 위한 비용이 지나치게 많이 소모된다는 단점이 존재한다.

[0074] 따라서 최근의 텐더민트나 하이퍼레저 패브릭 같은 블록체인 플랫폼은 작업증명을 통한 블록 생성 대신, 전체 네트워크 사용자의 2/3 이상이 정직하다면, 비-비잔틴(Non-Byzantine) 합의의 신뢰성이 보장되는 비잔틴 장애 허용(Byzantine fault tolerance, BFT) 계열의 합의 알고리즘을 사용하고 있다.

[0075] BFT 합의 알고리즘은 대부분 리더라고 불리는 프라이머리(Primary) 노드가 백업(Backup) 노드에 블록을 전송하고, 전체 노드가 해당 블록에 대한 합의 과정을 통해 블록 데이터의 신뢰성을 보장하는 방식을 채택한다. BFT 합의 알고리즘은 작업증명 방식보다 블록체인 플랫폼을 유지하는 비용이 적다는 장점이 있다. 현재의 합의 알고리즘에 따라 메시지 수의 오버헤드나 합의 과정의 속도 등이 달라지는데 이를 개선하여 최적의 속도를 내는 BFT 합의 알고리즘에 관한 연구가 활발히 진행 중이다. 이는 합의 알고리즘의 성능이 블록체인 성능 확장성에 가장 큰 영향을 주므로, 각 블록체인 플랫폼마다 고유의 합의 알고리즘을 적용하기 위한 것이다.

[0076] 전술한 바와 같이 비트코인이나 이더리움에서의 작업 증명(Proof-of-Work PoW) 알고리즘, 텐더민트나 하이퍼레저 패브릭의 경우, BFT 합의 알고리즘이나 권위 증명(Proof-of-Authority, PoA) 알고리즘 등 다양한 합의 알고리즘들을 볼 수 있다.

[0077] 이에 본 실시예의 블록 처리 기술에서는, 작업 증명(PoW)과 같이 블록 생성과 합의 과정이 통합되어 있는 경우는 고려하지 않으나, BFT 합의 혹은 권위 증명(PoA) 등의 알고리즘에서와 같이 블록 생성 단계와 합의 단계가 별도의 과정으로 정의되어 있는 경우, 즉 블록 생성 및 전송 단계와 합의 단계를 중첩시켜 블록체인 플랫폼의 성능을 개선하고자 한다.

[0078] 즉, 블록체인 플랫폼 합의 과정은 블록 전송 단계, 합의 단계, 결과 전파 단계의 3 단계들이 순차적으로 진행된다. 따라서, 블록의 모든 데이터를 수신한 후 합의 단계를 진행한다는 점에서, 특정 경우에, 빠른 네트워크 대역폭을 가진 노드들은 네트워크 대역폭이 가장 낮은 노드들이 합의 단계를 종료할 때까지 기다려야 하고, 이 때문에 블록체인 성능 확장성이 제한될 수 있다.

[0079] 특히, 클레이튼 블록체인 플랫폼에서 1초에 하나의 블록을 처리하는 것을 보장하고, 메타디움 블록체인 플랫폼에서 1초에 하나의 블록 처리를 권장 사항으로 규정하고 있는 것과 같이 블록체인 플랫폼이 초당 블록처리율을 보장해야 하는 경우, 블록 합의 단계에 탈중앙화 요소를 제한하게 될 수 있다.

[0080] 다시 말해서, 블록 전송, 합의, 결과 전파의 3개의 단계들이 순차적으로 진행될 때, 전체 블록 처리의 허용 시간을 1초로 규정한다면, 블록 합의 단계에 대한 가용 시간이 블록 전송 시간과 결과 전파 시간을 제외한 나머지 시간으로 제한된다. 이는 블록 합의 단계에 채택할 수 있는 합의 알고리즘의 선택을 제한하거나, 합의 단계에 참여하는 블록체인 노드들이 네트워크 상에서의 지리학적 위치들(geographical locations)이 서로 가깝게 배치되어야 하는 제한을 적용해야 하므로 결과적으로 탈중앙화 요소를 약화시키는 원인 중 하나가 될 수 있다.

[0081] 이러한 문제를 고려하여, 본 실시예에서는 초당 블록처리율을 보장하는 블록체인 플랫폼에서 하나의 블록을 처리하는데 필요한 3개의 순차적인 단계들 중 일부를 중첩시켜 합의 과정의 탈중앙화 특성을 높이고자 한다. 즉, 본 실시예에서는 순차적으로 진행되는 3개의 블록 처리 단계들 중 블록 전송 단계와 합의 단계를 중첩(overlapping)하는 방법을 이용한다.

[0082] 블록 전송 단계와 합의 단계를 중첩시키기 위해, 블록 일부 데이터 예컨대 해쉬값을 기반으로 블록 합의 단계를 시작하고, 블록 합의 단계를 진행하는 중에 해당 블록의 블록 데이터 전체를 수신하는 블록 전송 단계를 처리하도록 구성된다. 특히, 합의 단계를 블록 가용성(availability) 문제로 정의하여 블록 가용성 문제에 대한 해결 방안을 포함하는 블록 처리 기법을 제공할 수 있다.

[0084] 도 5는 본 발명의 다른 실시예에 따른 블록 처리 방법을 채용할 수 있는 블록체인 시스템에 대한 개념도이다. 그리고 도 6은 도 5의 블록체인 시스템에 채용할 수 있는 블록 생성자(block service provider, BSP)의 구성을 예시한 블록도이다.

[0085] 도 5를 참조하면, 블록체인 시스템(Blockchain System, 1000)은 클라이언트(Client, 100), 블록체인 서비스 제공자(Blockchain Service Provider, BSP, 200) 및 감사 네트워크(Audit Network, 300)를 포함한다. 블록체인 시스템(1000)은 블록 생성자(200)와 O(n) 비잔틴 장애 허용(Byzantine Fault Torence, BFT) 합의에 기반하는 감사 네트워크(300)에 의해 블록체인 서비스를 제공하는 고성능 허가형 블록체인 플랫폼을 구성할 수 있다. BSP(200)와 감사 네트워크(300)의 조합은 블록체인 네트워크로 지칭될 수 있다.

[0086] 또한, 본 실시예의 블록 처리 방법을 구현하는 블랙체인 시스템(1000)은 블록체인 서비스의 일종인 감사 체인(audit chain)를 이용하는 감사체인 플랫폼을 포함할 수 있다. 감사체인에서는 클라이언트(100)로부터의 트랜잭션들을 BSP(200)가 블록 데이터의 형태로 생성할 수 있다.

[0087] 각 구성요소를 좀더 구체적으로 설명하면, 클라이언트(100)는 클라이언트 게이트웨이(Client Gateway)로 지칭될

수 있고, 네트워크를 통해 사용자(users) 또는 사용자가 휴대하거나 사용하는 사용자 단말과 연결될 수 있다. 사용자 단말은 휴대 단말 또는 모바일 단말을 포함하거나 데스크탑 컴퓨터 등의 컴퓨팅 장치를 포함할 수 있다.

[0088] 사용자 단말은 넓은 의미에서 퍼스널 컴퓨터, 웹 서버, 컴퓨팅 서버, 애플리케이션 서버, 데이터베이스 서버, 파일 서버, 게임 서버, 메일 서버, 프록시 서버 또는 이들의 조합 형태를 포함할 수 있으며, 각 장치나 서버의 전부 또는 일부의 기능을 포함하도록 구성될 수 있다.

[0089] 또한, 사용자 단말은 무선 사용자 단말, 유선 사용자 단말 또는 이들의 혼합 형태인 유무선 사용자 단말을 포함할 수 있다. 무선 사용자 단말은 이동 단말(mobile terminal), 이동국(mobile station), 진보된 이동국(advanced mobile station), 고신뢰성 이동국(high reliability mobile station), 가입자국(subscriber station), 휴대 가입자국(portable subscriber station), 접근 단말(access terminal), 사용자 장비(user equipment) 등으로 지칭될 수 있고, 각 장치의 전부 또는 일부의 기능을 포함할 수 있다. 그리고 유선 사용자 단말은 네트워크에 연결되어 블록체인 시스템(1000)과 신호 및 데이터를 송수신할 수 있는 모든 단말 장치, 네트워크 터미널, 컴퓨팅 장치를 포함할 수 있다.

[0090] 클라이언트(100)는 호출(Invoke) 인터페이스를 통해 사용자 단말로부터 스마트 컨트랙트 실행을 위해 필요한 데이터를 수신할 수 있다. 호출 인터페이스에 접속한 사용자 단말은 블록체인 시스템(1000)의 블록체인 서비스에 참여한 상태가 될 수 있다. 클라이언트(100)는 블록 생성을 담당하는 BSP(200)에게 클라이언트 트랜잭션을 전달한다(S20). 클라이언트(100)는 클라이언트 SDK(Software Development Kit)(110) 및 취합기(Aggregator, 130)를 포함할 수 있다.

[0091] 클라이언트 SDK(110)는 클라이언트 트랜잭션을 BSP(200)로 전달할 수 있다. 클라이언트 SDK(110)는 간단히 SDK 또는 SDK 모듈로 지칭될 수 있다. SDK(110)는 사용자 단말로부터 호출(invoke) 인터페이스를 통해 수신한 스마트 컨트랙트 실행 인자들을 트랜잭션 제출 형태로 변환하고 이를 블록체인 네트워크의 BSP(200)에 제출할 수 있다.

[0092] 취합기(130)는 제출한 트랜잭션에 대한 응답을 블록체인 네트워크로부터 수신할 수 있다(S26, S28). 여기서, 응답은 BSP(200)가 트랜잭션을 실행한 직후에 전송하는 이벤트, 그리고 감사 네트워크(300)가 해당 트랜잭션에 대해 1차 검증과 2차 검증 직후에 전송하는 이벤트를 포함할 수 있다. 취합기(130)는 이벤트 취합기 또는 취합기 모듈로도 지칭될 수 있다.

[0093] 취합기(130)는 후술할 BSP(200)의 액티브 BSP(210)와 감사 네트워크(300)의 감사자들로부터 클라이언트 트랜잭션의 처리 결과를 이벤트의 형태로 취합할 수 있다. 이때 트랜잭션의 합의 수준은 필요에 따라 동적으로 결정될 수 있는데, 취합기(130)는 필요시 설정되는 트랜잭션 합의 수준에 따라 결정되는 트랜잭션 응답 결과를 SDK(110)을 통해 사용자 단말로 트랜잭션 처리 결과를 통보할 수 있다.

[0094] 즉, 사용자 단말은 클라이언트(100)를 경유하여 BSP(200)로 트랜잭션을 제출하고, 감사 네트워크(300)로부터 수신한 응답에 기반하여 트랜잭션 합의 결과를 확인할 수 있다. 이때 트랜잭션 합의 결과를 도출하는 합의 수준이 동적으로 변경 가능하며, 이는 블록체인 시스템(1000)의 안전성을 해하지 않는 수준에서 블록체인 서비스의 성능에 기반하여 합리적인 수준으로 결정될 수 있다.

[0095] BSP(200)는 블록체인 네트워크에서 블록 생성을 담당할 수 있다. BSP(200)는 BSP 서버들로서 액티브(active) BSP(210)와 스탠바이(stand-by) BSP(230)를 구비할 수 있다. 액티브 BSP는 활성 BSP로 지칭될 수 있고, 스탠바이 BSP는 대기 BSP로 지칭될 수 있다.

[0096] 즉, 액티브 BSP(210)는 SDK(110)로부터 수신한 트랜잭션을 상태 DB(270)에 저장되어 있는 스마트 컨트랙트(250)를 통해 실행하고, 그 결과를 포함한 블록(292)을 만들고 이를 블록체인(290)에 기록할 수 있다.

[0097] 액티브 BSP(210)에서 생성된 블록(292)은 액티브 BSP(210)에서 P2P(Peer to Peer) 오버레이 네트워크(overlay network, 400)를 통해 감사 네트워크(300)의 감사자들에게 전달될 수 있다(S22). P2P 오버레인 네트워크는 네트워크 상에서 피어들인 노드들이 서버의 도움없이 다른 피어들과 직접 정보를 공유하고 교환할 수 있도록 구성된 네트워크를 지칭할 수 있다. P2P 오버레인 네트워크는 간략히 P2P 네트워크로도 지칭될 수 있다.

[0098] 전술한 액티브 BSP(210)는 도 6에 도시한 바와 같이 시퀀서(211), 실행기(213), 블록생성기(215), 해시추출기(217) 및 해시&블록전파기(219) 모듈들을 구비할 수 있다. 액티브 BSP(210)는 상태 데이터베이스(database, DB, 270) 등의 저장소에 저장되는 스마트 컨트랙트(smart contract, 250)에 기초하여, 일정 시간 진행된 거래내용인 클라이언트 트랜잭션을 묶어 블록 데이터 형태의 블록(292)으로 생성하고 생성된 블록(292)을 기생성된 다른 블

록과 체인 구조로 연결하여 블록체인(290)을 생성할 수 있다. 전술한 저장소는 외부 저장소를 포함할 수 있으나, 이에 한정되지는 않는다.

[0099] 전술한 BSP(200)에서, 액티브 BSP(210)는 블록체인 트랜젝션 요청을 수신하고, 시퀀서 모듈(211)을 사용하여 트랜젝션의 실행 순서를 결정할 수 있다. 클라이언트 트랜젝션의 실행 순서가 결정되면, 액티브 BSP(210)는 실행기 모듈(213)을 사용하여 트랜젝션에 명시된 스마트 컨트랙트(250)를 실행하고 그에 대한 상태 DB(270)를 갱신할 수 있다.

[0100] 스마트 컨트랙트(250)의 실행 결과는 읽기 및 쓰기 집합을 포함하고, 이는 키-값 쌍으로 구성될 수 있다. 액티브 BSP(210)는 스마트 컨트랙트를 기반으로 하는 실행 내부 모듈(미도시)을 사용하여 클라이언트 트랜젝션들을 처리하고 그 결과를 블록생성기 모듈(215)을 통해 블록(510)으로 만들 수 있다. 이때, 해시추출기 모듈(217)은 블록생성기 모듈(215)의 블록 생성 과정에서 얻은 블록 해쉬를 추출할 수 있다. 물론, 해시추출기 모듈(217)은 해시값만을 추출할 수 있으나, 이에 한정되지 않고, 블록체인 플랫폼마다 다른 블록체인 블록의 블록 헤더 정보를 추출하도록 구성될 수 있다. 블록 헤더 정보에는 적어도 블록 해시가 포함된다. 그런 다음, 액티브 BSP(210)의 해시&블록전파기 모듈(219)는 P2P 네트워크(400)를 통해 블록 해시 또는 블록 헤더 정보를 감사 네트워크(300)의 감사자들에게 먼저 전파하고, 그 다음에 생성된 블록(292)을 P2P 네트워크(400)를 통해 감사 네트워트(300)의 감사자들로 추가로 전파할 수 있다.

[0101] 다시 도 5를 참조하면, 감사 네트워크(300)는 허가형 블록체인 네트워크에서 블록 처리 단계들 중 블록 합의 및 블록 전파를 담당할 수 있다. 블록 전파는 블록 합의 결과를 전파하는 결과 전파에 대응할 수 있다. 감사 네트워크(300)는 복수의 감사자들(Auditors)로 구성될 수 있다. 각 감사자는 감사자 노드(Aditor node 또는 A-node)로 지칭될 수 있다. 본 실시예에서 복수의 감사자들이 5개의 감사자들(310, 320, 330, 340, 350)인 것으로 설명하나, 이에 한정되지는 않고 더 적은 개수의 감사자들 혹은 더 많은 개수의 감사자들이 이용될 수 있다.

[0102] 감사 네트워크(300)의 감사자들은 프라이머리 감사자(이하 간략히 프라이머리(Primary) 또는 제1 감사자라고도 한다)를 중심으로 O(n) BFT 합의 프로토콜을 통해 블록 해시값에 대해 감사자들 간 블록 합의를 진행할 수 있다.

[0103] 감사 네트워크(300)는 감사 트랜잭션을 P2P 네트워크(400)를 통해 BSP(200)로 전송할 수 있다(S24). 감사 네트워크(300)는 블록 합의 결과에 따른 이벤트(Events)를 클라이언트(100)로 전송할 수 있다. 또한, 감사 네트워크(300)는 블록 합의 과정을 통해 생성되는 감사 블록을 클라이언트(100)의 취합기(130)로 전송할 수 있다(S28).

[0104] 감사 블록은 블록체인(290)의 블록(292)에 결합될 수 있다. 즉, 감사 네트워크(300)의 각 감사자는 BSP(200)로부터 받은 블록(292)으로 구성된 블록체인(290)과 감사자들이 생성하는 감사 블록으로 구성된 감사 체인을 통합하여 지역적으로(local) 관리할 수 있다. 다시 말해서, 각 감사자는 블록체인(290) 내 특정 블록(292)과 대응되는 동일한 높이의 감사 블록을 생성할 수 있다. 감사 블록은 합의 과정 동안 주고받은 메시지가 담겨 있어 사후에 감사 블록을 통해 합의 결과를 확인 및 증명하는데 이용될 수 있다.

[0105] 이러한 감사 블록들로 이루어진 감사 체인은 비잔틴 장애를 허용하고 장애가 존재하더라도 정상적인 시스템 동작이 가능하도록 비잔틴 장애에 대응하는데 사용될 수 있다. 즉, 감사 네트워크(300)의 전체 크기 즉, 전체 감사자들의 개수(n)가 3f+1이라고 할 때, 감사 네트워크(300)의 전체 감사자들은 최대 f개까지의 비잔틴 감사자들을 포함하면서 정상적으로 동작할 수 있다.

[0106] 전술한 감사 네트워크(300)의 감사자들은 서로 연결되어 적어도 하나 이상의 네트워크를 형성할 수 있다. 각 감사자는 감사 네트워크(300) 내에 연결되는 다른 감사자와 독립적으로 감사 트랜젝션을 제출하고 분석할 수 있다. 각 감사자는 생성된 블록을 P2P 네트워크(400)를 통해 다른 감사자들에게 전파할 수 있다. 각 감사자는 수신한 블록에 대해 그 요약 정보를 담은 감사 트랜젝션을 생성하고 생성한 감사 트랜젝션을 BSP(200)로 제출할 수 있다. 구현에 따라서, 감사자는 수신한 블록에 대한 감사 트랜젝션을 생성할 수 있으나, 감사를 위한 별도의 블록체인을 반드시 생성할 필요는 없다.

[0107] 전술한 블록체인 시스템(1000)에서는 BSP(200)에서 블록체인 연산의 상당 부분과 블록 생성 단계를 담당하므로, 감사 체인의 구성에 의하면, 사용자 단말의 연산 부담이 적고 전체적인 블록체인 서비스 시간 지연이 단축되며 성능이 향상될 수 있다. 그리고, BSP(200)와 분리된 감사 네트워크(300)를 통해 비잔틴 BSP를 탐지하여 악의적인 공격에 대응할 수 있다. 비잔틴 BSP는 BSP(200)가 악의적 공격자에게 탈취된 상태이거나, BSP(200)의 운영 주체가 악의적인 의도를 가진 상태로 이미 합의된 결과를 임의로 변조하려는 등의 위협적 행동을 일으키는 장치나 서버를 지칭할 수 있다.

[0108] 특히, 감사 네트워크(300)의 감사자들은 BSP(200)에 대한 감사 또는 감사를 수행하며, 적어도 하나의 네트워크를 형성하여 액티브 BSP(210)가 만든 블록(292)을 토대로 감사 트랜젝션을 상호 독립적으로 도출하여 비잔틴 BSP를 탐지할 수 있다. 이러한 감사자들은 BSP(200)와 독립적으로 기능하며 클라이언트(100)를 통해 전달되는 모든 트랜젝션을 감사할 수 있고, BSP(200)에 의하여 트랜젝션이 임의로 변경되는 행위 등의 악의적 행동을 탐지되면, 해당 BSP(200)의 액티브 서버(210)가 기능하지 못하도록 조치할 수 있다. 또한 액티브 BSP(210)의 악의적 행동이 탐지되면, 감사자들은 액티브 BSP(210)의 변경을 위한 BSP 변경 프로토콜을 개시할 수 있다.

[0109] 전술한 블록체인 시스템(1000)의 구성요소들인 통신 노드들 간 통신 구조를 좀더 구체적으로 설명하면, 클라이언트(100)는 내부적으로 실제 트랜젝션을 제출하는 SDK 모듈(110)과 BSP(200)와 감사 네트워크(300)로부터 이벤트를 수신하는 취합기 모듈(130)을 구비한다. SDK 모듈(110)이 트랜젝션을 현재의 액티브 BSP(210)로 제출하면, 액티브 BSP(210)는 실행기 모듈(213)을 통해 트랜젝션을 실행하고 그 결과로 블록(510)을 만들어서 감사 네트워크(300)의 감사자들에게 전파할 수 있다. 이와 동시에 액티브 BSP(210)는 트랜젝션 처리 완료 이벤트 예컨대, 'Spec' 이벤트를 취합기(130)로 전송할 수 있다. 'Spec' 이벤트는 일정 수 이상의 감사자들이 합의에 동의한 커밋(commint) 이벤트의 일종일 수 있다.

[0110] 감사 네트워크(300)의 감사자들은 블록(292)을 수신하고, 수신한 블록(292)에 대한 1차 검증 과정을 수행 후 문제가 없으면, 해당 블록 내 트랜젝션을 제출한 클라이언트에게 1차 검증 완료 이벤트 예컨대, 'Ordered' 이벤트를 취합기(130)로 전송할 수 있다. 'Ordered' 이벤트는 일정 수 이상의 감사자들이 합의에 동의한 커밋(commint) 이벤트의 일종으로서, 'Ordered' 커밋 이벤트로 지칭될 수 있다.

[0111] 또한, 감사 네트워크(300)의 감사자들은, 블록 내 감사용 트랜젝션이 있을 경우, 감사용 트랜젝션을 추출하여 비잔틴 장애 허용(Byzantine Fault Tolerance, BFT) 프로토콜 기반으로 합의 단계를 수행할 수 있다. 합의 결과에 문제가 없으면, 감사자들은 합의된 블록 내 트랜젝션을 제출한 클라이언트에게 'Commit' 이벤트를 취합기(130)로 전송할 수 있다.

[0112] 전술한 감사 네트워크(300)를 사용하면, 블록 생성은 BSP(200)에서 담당하므로, 감사자들 간의 비잔틴 합의를 도출하는데 소요되는 시간을 단축할 수 있고, 감사자들 간의 비잔틴 합의 도출 과정에서 메시지 통신 패턴을 선형화할 수 있고, 블록 해시 정보 기반으로 분석 비용을 경량화할 수 있어 블록 합의 과정의 효율을 크게 향상시킬 수 있다. 비잔틴 합의 도출 과정에서 감사자들 간의 메시지 통신 패턴을 선형화하는 것은 감사자들의 수에 따라 메시지 통신 패턴을 선형화할 수 있다는 의미에서 O(n) BFT 합의로 지칭될 수 있다.

[0113] 블록체인 네트워크에서 블록 생성 및 블록 합의가 분리 처리된 후, 'commit' 여부가 이벤트(커밋 이벤트) 형태로 취합기(130)로 수신될 수 있다. BSP(200) 및 감사자들로부터 수신된 이벤트들은 취합기 모듈(130)에서 취합된 후 커밋 이벤트에 기반하여 블록체인 서비스의 정책에 맞추어 트랜젝션 합의 여부가 결정될 수 있다. 트랜젝션 합의 여부가 취합기 모듈(130)로부터 블록체인 서비스의 정책에 맞추어 최종적으로 사용자 단말에 통지됨으로써 사용자 트랜젝션이 처리 종료될 수 있다.

[0114] 특히, 감사 네트워크(300)의 감사자들은 액티브 BSP(210)으로부터 블록 해시 또는 블록 헤더 정보를 받고, 해당 블록의 모든 데이터를 받기 전에 블록 합의 단계를 먼저 시작하고, 나중에 블록의 모든 데이터가 수신될 때 해당 블록 내 정보를 통해 합의 단계를 완료하도록 구성될 수 있다. 이때, 감사 네트워크(300)의 각 감사자는 먼저 수신한 블록 해시의 해시값과 나중에 수신된 블록 내 해시값을 토대로 해당 블록 해시나 블록의 유효성을 검증할 수 있다.

[0115] 한편 사용자 단말에서 유효하지 않은 트랜젝션이 생성되는 경우, 블록체인 시스템(1000)은 해당 트랙젝션을 블록체인 네트워크에서 처리하지 않고 폐기(abort)할 수 있다.

[0116] 도 7은 도 5의 블록체인 시스템에 적용가능한 비교예의 블록 처리 과정을 설명하기 위한 도면이다.

[0117] 도 7을 참조하면, 비교예의 블록 처리 장치는, 블록 전송과 블록 합의에 대하여 중첩(overlapping) 기법을 적용하지 않은 경우로서, 감사체인 기반으로 구성되고 동작할 수 있다.

[0118] 즉, 비교예에서 액티브 BSP인 BSP 서버는 블록(B_h)를 생성하고(S71), 생성한 블록(B_h)을 감사 네트워크의 A-노드들(A-nodes)로 전송한다(S73). 블록 전송 단계(S73)는 BSP의 입장에서 BSP가 감사 네트워크에 대해 블록체인의 블록을 제안(proposal)하는 과정에 대응하고, 감사 네트워크의 입장에서는 합의 과정에 대한 준비(prepare) 단계 혹은 사전 준비(pre-prepare) 단계에 대응할 수 있다.

[0119] 본 비교예에서 감사 네트워크의 A-노드들은 제1 A-노드 내지 제4 A-노드를 포함하는 것으로 설명하기로 한다.

여기서 제1 A-노드, 제2 A-노드, 제3 A-노드 및 제4 A-노드는 기재된 순서대로 A1, A2, A3 및 A4로 각각 지칭될 수 있다. 그 중에 A1은 프라이머리에 해당할 수 있다.

[0120] 블록(B_h)을 수신한 A-노드들(A1 내지 A4)은 해당 블록에 대한 합의를 이루고 각자의 감사체인에 블록을 추가한다. 이때, A-노드들에서 합의가 시작되기 위해서는 블록 전송 단계(S73)에서 BSP 서버가 블록 데이터 즉, 블록의 모든 데이터를 전송하고 각 A-노드가 블록의 모든 데이터를 수신하는 것을 완료해야만 한다.

[0121] 즉, 감사 네트워크의 감사자들(A1, A2, A3, A4)은 BSP 서버로부터 블록체인을 위한 블록(B_h)을 받고, 감사 트랜잭션을 제1 A-노드(A1)인 프라이머리(Primary)로 전달하고, 제1 A-노드(A1)로부터 쿼럼인증서(quorum certificate, QC)를 받는 과정을 반복하며 합의를 위한 준비(prepare)와 동의(commit) 과정을 진행할 수 있다. 여기서, 감사 네트워크 내에서 감사자들 간에 주고받는 메시지에 관한 O(n) BFT 합의 프로토콜에는 감사 트랜잭션(Audit Transaction, audit TX)과 쿼럼인증서(Quorum Certificate, QC)가 포함될 수 있다.

[0122] 여기서, 감사 트랜잭션은 제2 A-노드 내지 제4 A-노드(A2, A3, A4) 각각에서 제1 A-노드(A1)로 투표(vote) 형식으로 전달될 수 있다. 쿼럼인증서는 1차 검증 단계인 준비(prepare) 단계에서의 커밋(commit)에 의해 생성되는 준비 QC(prepare QC)와, 2차 검증 단계인 커밋(commint) 단계에서의 결정(decide)에 의해 생성되는 커밋 QC(commit QC)를 포함할 수 있다. 그리고, 감사 네트워크의 각 감사자에 의해 생성되는 감사 블록은 해당 블록에 대한 블록 해시값을 포함할 수 있다.

[0123] 한편, 감사 네트워크의 감사자들(A1 내지 A4)에 의하여 수행되는 1차 검증의 동작 흐름을 예시하면 다음과 같다.

[0124] 먼저, BSP 서버로부터 P2P 네트워크를 통해 블록을 수신하면, 감사자들은 블록 단위 검증을 수행할 수 있다. 블록 단위 검증은 블록 서명 검증, 해시 일관성 검증 등을 포함할 수 있다.

[0125] 다음, 감사자들은 블록 내 트랜잭션들을 순회하면서 트랜잭션 단위 검증을 수행할 수 있다. 트랜잭션 단위 검증은 클라이언트 서명 검증, 트랜잭션 유효성 검증, 트랜잭션 실행 무결성 검증을 포함할 수 있다.

[0126] 여기서, 클라이언트 서명 검증은, 해당 트랜잭션을 제출할 때의 클라이언트가 제공한 스마트 컨트랙트 함수 및 그 인자 정보, 타임스탬프 등의 클라이언트가 생성한 고유의 정보에 대한 위변조 여부를 서명 검증을 통해 확인하는 것을 의미할 수 있다.

[0127] 트랜잭션 유효성 검증은 트랜잭션 ID 구성 유효성, 트랜잭션의 널 필드 검사, 트랜잭션 제출 및 수행 권한 검증 등을 포함할 수 있다.

[0128] 트랜잭션 실행 무결성 검증은, 감사자들에 의해, BSP가 클라이언트로 요청한 스마트 컨트랙트를 올바르게 실행했는지 여부를 검증하는 것을 포함할 수 있다. 즉, 스마트 컨트랙트 실행 결과가 주어진 입력 값에 결정적으로 계산이 되는지 검증할 수 있다. 예를 들어, 특정 스마트 컨트랙트 함수 실행에 제공된 인자 값, 그리고 BSP가 상태 데이터베이스로부터 추가로 제공한 읽기 집합으로부터 계산된 결과가 실제 쓰기 집합 내의 값과 동일한지 검증하는 단계를 포함할 수 있다. 검증 결과에 문제가 없으면, 감사자들은 'Ordered' 커밋 이벤트를 블록 내 트랜잭션을 제출한 클라이언트에 전송할 수 있다. 또한, 액티브 BSP가 악의적 행위를 하고 이를 감사자들 중 적어도 하나가 탐지하면, 감사자들은 BSP 변경 프로토콜을 시작하고, 그 결과로 대기 중인 스텐바이 BSP들(도 5의 230 참조) 중 어느 하나의 BSP가 현재의 액티브 BSP(210)를 대체하도록 동작할 수 있다.

[0130] 도 8은 도 5의 블록체인 시스템에 적용할 수 있는 본 실시예의 블록 처리 과정을 설명하기 위한 도면이다.

[0131] 본 실시예의 블록 처리 과정은 중첩(overlapping)을 적용한 감사 체인의 구체적인 구성과 동작 형태로 설명될 수 있다.

[0132] 도 8을 참조하면, BSP 서버는 블록(B_h)를 생성하고(S81), 생성한 블록(B_h)의 해시인 H(B)를 먼저 A-노드들(A1, A2, A3, A4)로 전송한다(S82).

[0133] 블록(B_h)의 해시인 H(B)를 수신한 A-노드들(A1, A2, A3, A4)은 해당 블록 해시의 해시값으로 블록 합의를 진행한다(S85). 해시값만으로 진행되는 블록 합의는 블록 합의(해쉬) 또는 블록 해쉬 합의로 표현될 수 있고, 1차 검증과 2차 검증 단계들을 포함할 수 있다.

[0134] 블록(B_h)의 해시인 H(B)를 전송한 BSP 서버는 모든 해시값이 전송된 후 전체 블록을 A-노드들(A1, A2, A3, A4)로 전송할 수 있다. 즉, 도 8에서 블록 해시 전송 단계(S82) 및 블록 해쉬 합의 단계(S85)의 제1 블록 처리 과정와 블록 전송 단계(S83)를 포함한 제2 블록 처리 과정은 시간 흐름 상에서 서로 중첩되어 병렬적으로 진행될 수 있다.

[0135] 즉, A-노드들(A1 내지 A4)은 블록 해시인 H(B)라는 데이터로 합의를 진행하는 동시에 백그라운드에서는 전체 블록(B_h)에 대한 수신 작업을 병렬적으로 진행할 수 있다. 이와 같이, 본 실시예의 블록 처리 장치는 A-노드들 중 어느 하나에서 수행되는 블록 처리 방법으로서, 전술한 블록 해쉬 기반의 합의와 동시에 전체 블록 데이터를 수신하는 방법을 중첩(overlapping) 기법을 채용한다.

[0136] 블록 해쉬 합의(S85)가 종료된 후, 블록 전송 단계(S83)에서 이미 병렬적으로 수신된 블록(B_h)에 대해 A-노드들이 수신된 블록(B_h)의 해시값을 각각 계산하고, 계산된 해시값을 블록 해쉬 합의 단계(S85)에서 진행되었던 해시값 H(B)과 비교하여 유효한 블록임을 확인할 수 있다. 해당 블록(B_h)이 유효한 블록으로 확인되면 A-노드들은 각자의 감사체인에 해당 블록을 추가하고(S87), 이벤트(event) 메시지를 포함하는 합의 결과를 클라이언트 취합기(Aggregator)로 보내어 합의 단계를 종료할 수 있다.

[0137] 블록 해시값인 H(B)의 데이터 크기는 블록(B_h)의 데이터 크기와 상관없이 언제나 일정할 수 있다. 따라서 A-노드들(A1, A2, A3, A4)가 블록 해시를 전송받고 합의 진행을 시작하는 속도는, 기존의 중첩(overlapping) 기법이 적용되지 않은 블록체인이나 감사체인 기반의 플랫폼과 비교할 때 블록의 크기의 비례해서 증가할 수 있다.

[0138] 예를 들어, BSP 서버의 데이터 전송 속도를 10KB/s, 블록(B_h)의 크기를 100KB, 블록 해시값인 H(B)의 크기를 1KB라 가정하면, 비교예의 감사체인 플랫폼에서 BSP 서버가 4개의 A-노드들에게 블록(B_h)를 전송하는데 걸리는 시간은 40초(=400/10)이다. 즉 A-노드들은 BSP 서버가 블록 전송을 시작한 후 40초 이후에 블록에 대한 합의를 진행할 수 있다. 그러나, 중첩(overlapping) 기법을 적용하면, 블록 해시값만 전송한 후 블록 합의를 진행할 수 있다. 이 경우에, 감사체인 플랫폼에서 BSP 서버가 4개의 A-노드들(A1 내지 A4)에게 블록(B_h)를 전송하는데 걸리는 시간은 0.4초(4/10)로 위의 비교예의 감사체인 플랫폼의 경우보다 100배 더 빠른 속도로 A-노드들의 합의를 시작할 수 있다.

[0139] 특히, A-노드들(A1, A2, A3, A4)의 지정학적 위치가 지구 상에 물리적으로 멀리 떨어져 있는 경우, 예를 들어, 대한민국 서울, 미국 워싱턴, 영국 런던, 및 호주 시드니에 각각 위치하는 경우, 그리고 이들을 연결하는 P2P 네트워크가 지역에 따라 지연 시간이 다른 경우에도 각 A-노드의 합의 단계의 시작 시간이나 진행 시간을 거의 동일하게 할 수 있고, 그에 의해 기존 대비 블록체인 합의 과정에서의 탈중앙화를 크게 향상시킬 수 있다.

[0140] 도 9는 도 5의 블록체인 시스템에 적용할 수 있는 본 실시예의 비잔틴 BSP 대응 블록 처리 과정을 설명하기 위한 도면이다.

[0141] 본 실시예에서는 블록 처리 장치가, 중첩(overlapping)을 적용한 감사체인 플랫폼에서 악의적인 BSP 서버, 다시 말해서 비잔틴(Byzantine fault tolerance, BFT) BSP 서버의 라이브니스(Liveness) 공격 등의 무결성에 반하는 동작에 대응하고 결과적으로는 블록에 대한 가용성(availability)을 보장하는 과정을 포함하는 블록 처리 방법을 중심으로 설명하기로 한다. 여기서 라이브니스 공격은 사진, 동영상 또는 모형과 같은 매체물을 이용한 불법 인증 시도 또는 스푸닝(spoofing) 공격의 일종을 지칭할 수 있다.

[0142] 도 9를 참조하면, 악의적인 BSP 서버는 의도적으로 합의를 방해하기 위해 A-노드들(A1 내지 A4)로 블록 전송(S93 참조)을 수행하지 않아 결과적으로 블록 가용성(availability)를 보장하지 못하게 할 수 있다. 따라서 A-노드들(A1 내지 A4)은 블록 해쉬 H(B)를 수신받은 후, 자신의 타이머(900)를 제1 시간을 설정하여 작동시킬 수 있다.

[0143] 타이머(900)가 제1 시간에서 타임아웃이 되기 전에 블록(B_h)이 수신되면, A-노드들(A1 내지 A4)은 자신의 타이머(900)를 초기화할 수 있다. 그리고, A-노드들(A1 내지 A4)은 블록 해쉬 합의 단계(S95)에서 사용된 각 해쉬값과 블록 전송을 통해 수신한 블록의 해시값을 비교하여 블록의 유효성을 최종적으로 확인하여 블록 합의 단계(S97)를 완료할 수 있다.

[0144] 한편, 일정한 시간 즉, 제1 시간 내에 블록(B_h)을 수신하지 못하여 타이머(900)가 제1 시간을 지나면서 타임아

웃이 되면(S98), A-노드들(A1 내지 A4)은 해당 BSP 서버를 악의적인 동작을 수행하는 BSP 서버로 판단하고, BSP 교체 프로토콜(Change protocol)을 실행할 수 있다(S99).

[0145] BSP 교체 프로토콜에 의하면, 블록 처리 장치는 블록 생성자(BSP)의 현재 악의적인 BSP 서버로 판명된 액티브 BSP를 다른 BSP 서버인 스탠바이 BSP로 교체할 수 있다. 이와 같이, 본 실시예의 중첩(overlapping) 기법을 이용하넌 블록 처리 방법은 전술한 대응 방법으로 악의적인 BSP 서버의 라이브니스 공격 등의 악의적인 동작에 대응하고 그에 의해 결과적으로 블록 가용성(availability)를 보장할 수 있다.

[0146] 전술한 구성에 의하면, 중첩(overlapping) 기법이 적용되지 않은 블록 처리 과정(도 3 참조)와 중첩 기법이 적용된 블록 처리 과정(도 4 참조)의 블록 합의 단계에 대한 가용 시간에서 큰 차이가 있음을 알 수 있다. 즉, 도 3의 비교예에서와 같이, 목표 블록 처리 시간(T1)이 주어져 있을 때, 블록 합의에 대한 가용 시간은 블록 전송 시간과 결과 전파 시간에 제한을 받는다. 다시 말해서, 어떤 노드의 제한된 네트워크 대역폭으로 인해, 블록 전송이 늦게 완료될 경우, 블록 합의에 대한 가용 시간은 줄어들 수밖에 없다. 한편, 도 4의 본 실시예에서와 같이, 블록 해쉬 전송이 완료된 이후 블록 해쉬에 대한 합의를 시작하는 경우, 블록 해쉬 전송 과정은 네트워크 대역폭 소모량이 적은 블록 해시에 의해 동작하므로 해당 데이터를 수신하는 노드들 간의 수신 속도 차이가 비교예와 비교할 때 현저하게 줄어들어 합의 참여 노드 간의 블록 합의 시작이 비슷한 시점에서 이루어질 수 있다. 따라서 본 실시예에서는 블록 합의에 대한 가용 시간이 비교예에 비해 크게 되고, 블록 합의 단계에서의 합의 알고리즘 선택의 제한이나 합의 단계에 참여하는 블록체인 노드들의 네트워크 위치상 거리 제한을 완화할 수 있다. 결과적으로 중첩 기법을 적용한 블록체인 플랫폼은 합의 알고리즘이나 합의 참여 노드들의 네트워크 거리와 상관없이 일정한 블록 처리율을 보장할 수 있다.

[0147] 도 10은 본 발명의 또 다른 실시예에 따른 블록 처리 장치에 대한 개략적인 블록도이다.

[0148] 본 실시예에서 블록 처리 장치는 블록 생성자와 감사자를 포함한 블록체인에 참여하는 노드들(nodes) 중 어느 하나일 수 있다. 블록체인 노드는 클라이언트나 클라이언트 게이트웨이를 포함할 수 있고, 또한 블록 생성자나 감사자를 포함할 수 있다.

[0149] 도 10을 참조하면, 블록 처리 장치(1100)는 적어도 하나의 프로세서(1010), 메모리(1020) 및 네트워크와 연결되어 통신을 수행하는 송수신 장치(1030)를 구비할 수 있다. 또한, 블록 처리 장치(1100)는 입력 인터페이스 장치(1040), 출력 인터페이스 장치(1050), 저장 장치(1060) 등을 더 구비할 수 있다. 블록 처리 장치(1100)에 포함된 각각의 구성 요소들은 버스(bus, 1070)에 의해 연결되어 서로 통신을 수행할 수 있다.

[0150] 다만, 블록 처리 장치(1100)에 포함된 각각의 구성요소들은 공통 버스(1070)가 아니라, 프로세서(1010)를 중심으로 개별 인터페이스 또는 개별 버스를 통하여 연결될 수 있다. 예를 들어, 프로세서(1010)는 메모리(1020), 송수신 장치(1030), 입력 인터페이스 장치(1040), 출력 인터페이스 장치(1050) 및 저장 장치(1060) 중에서 적어도 하나와 전용 인터페이스를 통하여 연결될 수 있다.

[0151] 프로세서(1010)는 메모리(1020) 및 저장 장치(1060) 중에서 적어도 하나에 저장된 프로그램 명령(program command)을 실행할 수 있다. 프로그램 명령은 적어도 하나의 명령이나 적어도 하나의 소프트웨어 모듈을 포함할 수 있다. 이러한 프로세서(1010)는 중앙 처리 장치(central processing unit, CPU), 그래픽 처리 장치(graphics processing unit, GPU), 또는 본 발명의 실시예에 따른 방법들 중 적어도 하나의 방법이 수행되는 전용의 프로세서를 의미할 수 있다.

[0152] 메모리(1020) 및 저장 장치(1060) 각각은 휘발성 저장 매체 및 비휘발성 저장 매체 중에서 적어도 하나로 구성될 수 있다. 예를 들어, 메모리(1020)는 읽기 전용 메모리(read only memory, ROM) 및 랜덤 액세스 메모리(random access memory, RAM) 중에서 적어도 하나로 구성될 수 있다.

[0153] 송수신 장치(1030)는 네트워크를 통해 사용자 단말과의 연결을 지원하는 수단, 블록체인 시스템 내 다른 구성요소들과의 연결을 지원하는 수단, 또는 이러한 수단에 상응하는 기능을 수행하는 구성부를 포함할 수 있다. 송수신 장치(1030)는 P2P 네트워크를 지원하거나 유선, 무선 또는 유무선 네트워크를 지원하는 적어도 하나의 서브 통신시스템을 포함할 수 있다.

[0154] 입력 인터페이스 장치(1040)는 키보드, 마이크, 카메라, 터치패드, 터치스크린 등의 입력 수단들과, 입력 수단들 중에서 선택되는 적어도 하나를 통해 입력되는 신호를 기저장된 명령과 매핑하거나 기설정된 규칙에 따라 해석하여 프로세서(2100)로 전달하는 입력 신호 처리부를 포함할 수 있다.

[0155] 출력 인터페이스 장치(1050)는 프로세서(1010)의 제어에 따라 출력되는 신호를 기저장된 신호 형태나 레벨로 매

핑하거나 처리하는 출력 신호 처리부와, 출력 신호 처리부의 신호나 정보를 진동, 빛, 소리 등의 형태로 출력하는 출력 수단을 포함할 수 있다. 출력 수단은 스피커, 디스플레이 장치, 프린터, 광 출력 장치, 진동 출력 장치 등을 포함할 수 있다.

[0156] 또한, 전술한 블록 처리 장치(1100)는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소와 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 블록 처리 장치(1100)는, 프로세서, 컨트롤러, 마이크로컴퓨터, 마이크로프로세서, 디지털 신호 프로세서(Digital Signal Processor), FPA(Field Programmable Array), ALU(Arithmetic Logic Unit), PLU(Programmable Logic Unit), 또는 명령(Instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다.

[0157] 이러한 블록 처리 장치(1100)는 또한 운영 체제(Operating System, OS) 및 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 탑재할 수 있다. 그리고 블록 처리 장치(1100)는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수 있다. 이해의 편의를 위하여, 블록 처리 장치(1100)는 하나가 사용되는 것으로 설명된 경우도 있지만, 구현에 따라서 블록 처리 장치(1100)는 복수의 처리 요소들(Processing Elements) 및/또는 복수 유형의 처리 요소들을 포함할 수 있다. 예를 들어, 블록 처리 장치(1100)는 복수의 프로세서를 포함하거나 또는 하나의 프로세서와 하나의 컨트롤러를 포함할 수 있다. 또한, 블록 처리 장치(1100)는 병렬 프로세서(Parallel Processor)와 같은 처리 구성(Processing Configuration)을 포함하는 것도 가능하다.

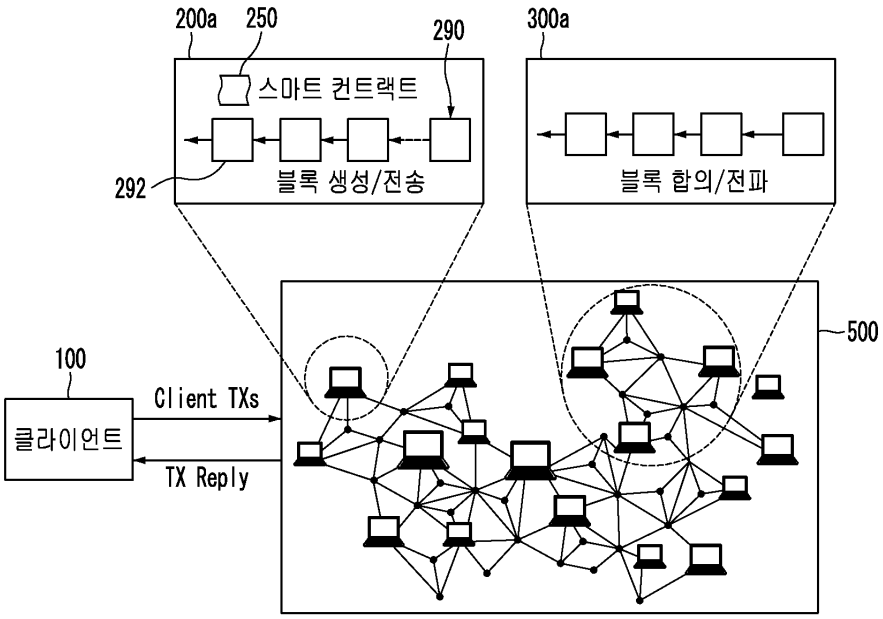
[0158] 본 발명에 따른 방법들은 다양한 컴퓨터 수단을 통해 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 컴퓨터 판독 가능 매체에 기록되는 프로그램 명령은 본 발명을 위해 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수 있다.

[0159] 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.

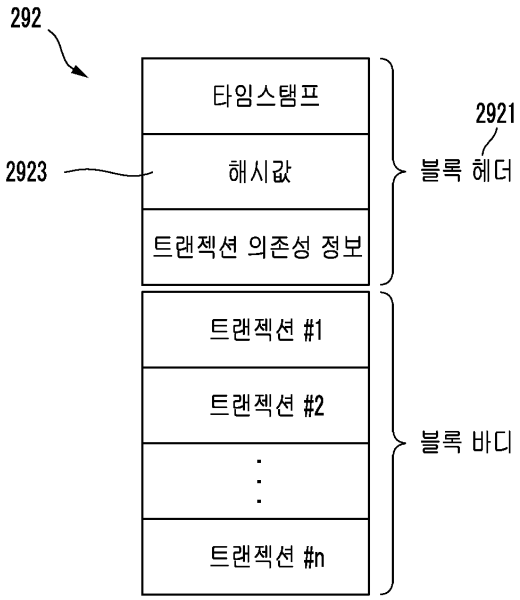
[0160] 이상 실시예를 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

도면

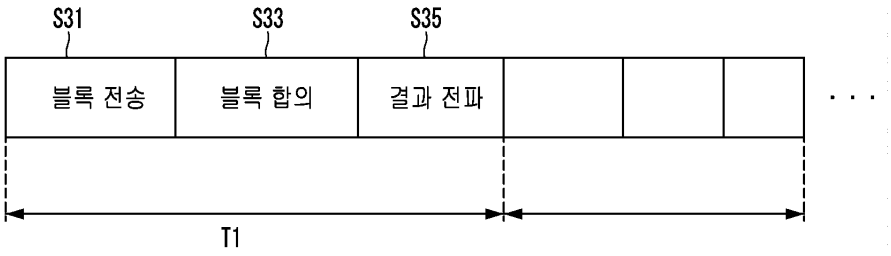
도면1



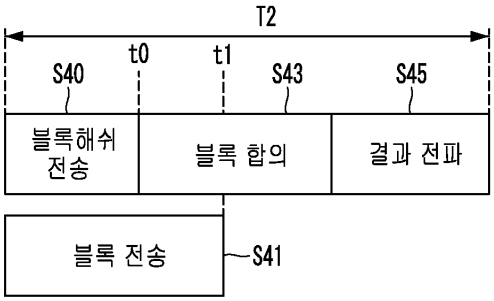
도면2



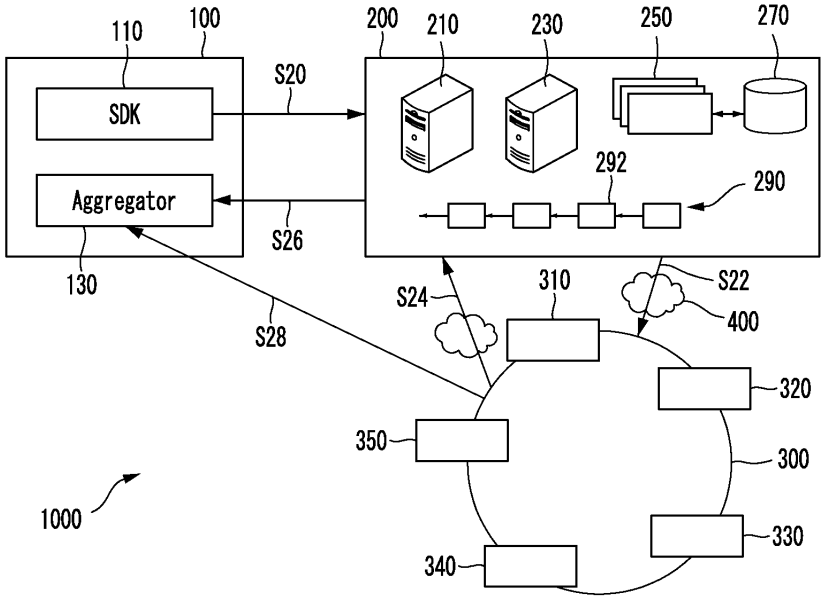
도면3



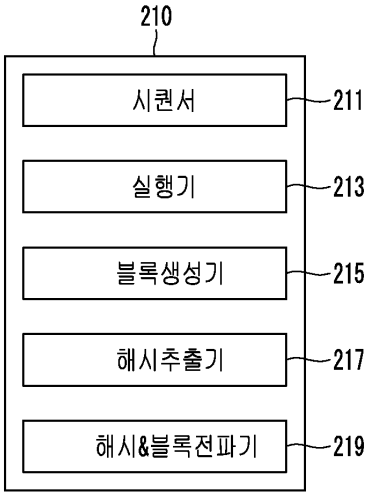
도면4


T2
t0
t1
S40
S43
S45
블록해쉬 전송
블록 합의
결과 전파
블록 전송
S41

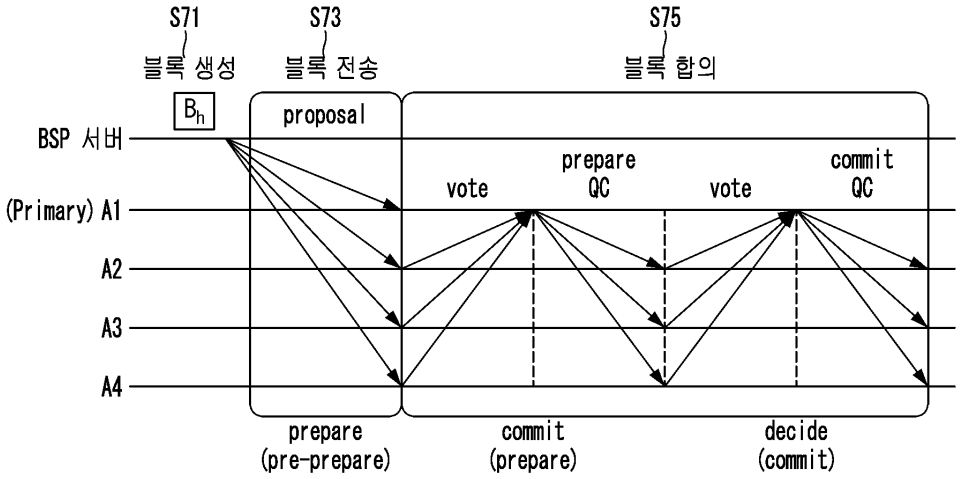
도면5


110
100
200
210
230
250
270
SDK
S20
Aggregator
S26
292
290
130
S28
S24
310
S22
400
320
350
300
330
340
1000

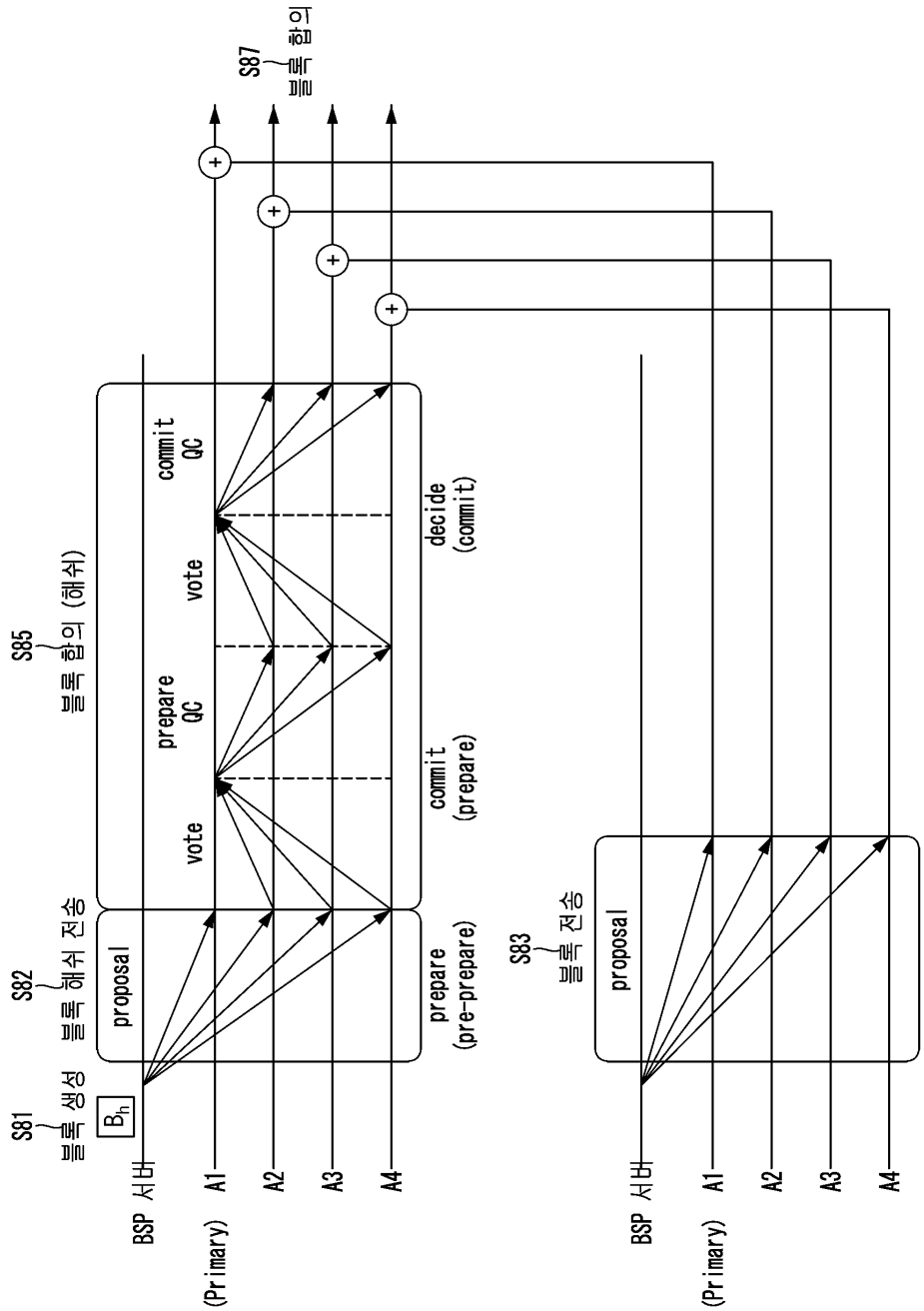
도면6


210
시퀀서
211
실행기
213
블록생성기
215
해시추출기
217
해시&블록전파기
219

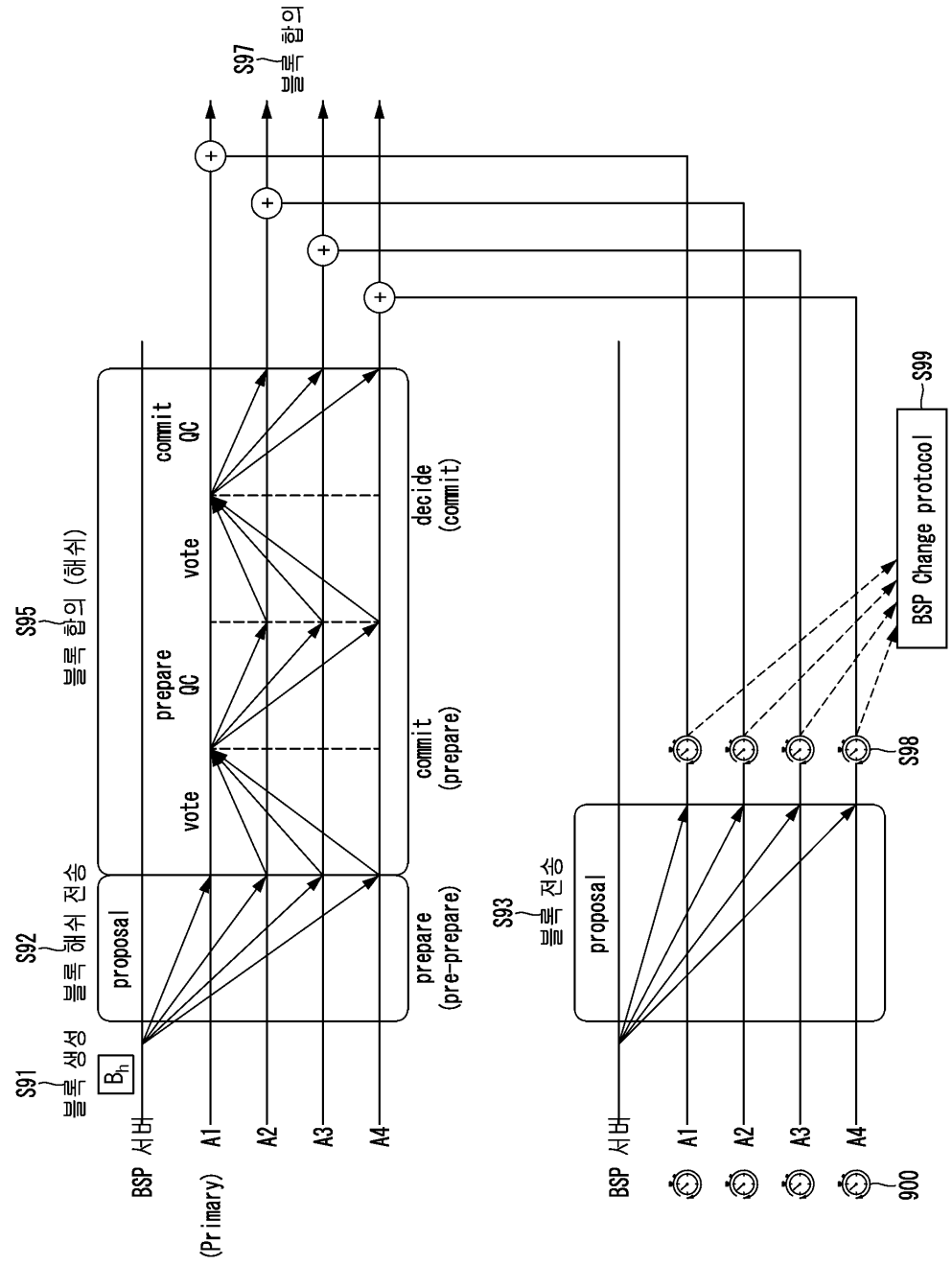
도면7


S71
블록 생성
S73
블록 전송
S75
블록 합의
B_h
proposal
BSP 서버
(Primary) A1
A2
A3
A4
vote
prepare QC
vote
commit QC
prepare (pre-prepare)
commit (prepare)
decide (commit)

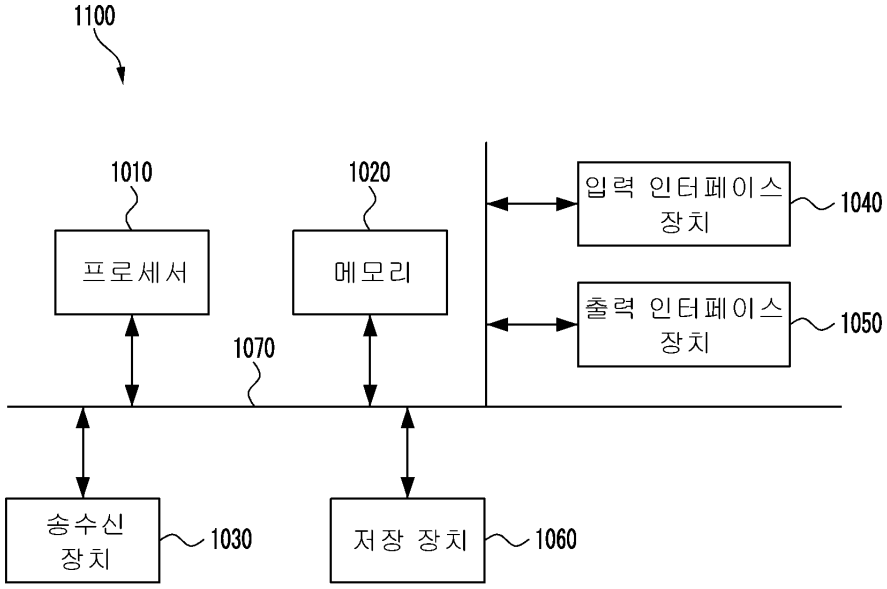
도면8


S81
블록 생성
B_h
BSP 서버
(Primary)
A1
A2
A3
A4
S82
블록 해쉬 전송
proposal
prepare
(pre-prepare)
vote
prepare QC
commit
(prepare)
S85
블록 합의 (해쉬)
vote
commit QC
decide
(commit)
S87
블록 합의
S83
블록 전송
proposal

도면9


S91
블록 생성
B_h
S92
블록 해쉬 전송
proposal
S95
블록 합의 (해쉬)
prepare QC
vote
commit QC
vote
prepare (pre-prepare)
commit (prepare)
decide (commit)
S97
블록 합의
BSP 서버
(Primary)
A1
A2
A3
A4
S93
블록 전송
proposal
BSP 서버
A1
A2
A3
A4
900
S98
S99
BSP Change protocol

도면10


1100
1010
프로세서
1020
메모리
입력 인터페이스 장치
1040
출력 인터페이스 장치
1050
1070
송수신 장치
1030
저장 장치
1060