

(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국

(43) 국제공개일
2020년 1월 2일 (02.01.2020)



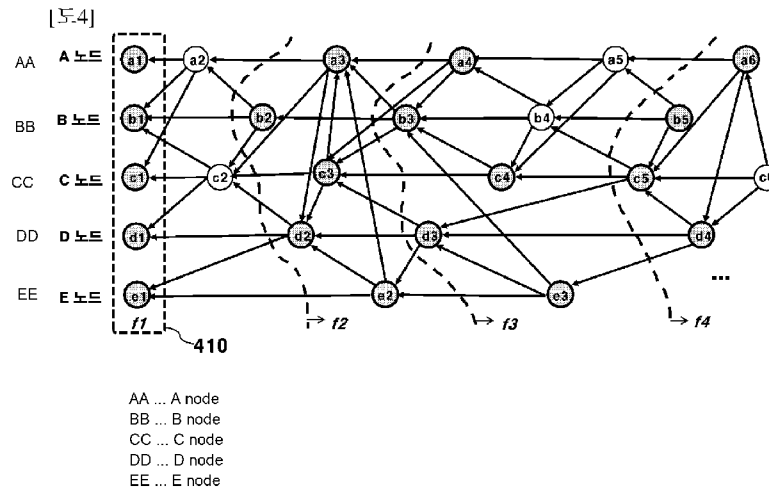
(10) 국제공개번호
WO 2020/004756 A1

- (51) 국제특허분류:
H04L 29/06 (2006.01) *H04L 9/06* (2006.01)
- (21) 국제출원번호: PCT/KR2019/000425
- (22) 국제출원일: 2019년 1월 11일 (11.01.2019)
- (25) 출원언어: 한국어
- (26) 공개언어: 한국어
- (30) 우선권정보:
10-2018-0073882 2018년 6월 27일 (27.06.2018) KR
- (71) 출원인: 팬텀 파운데이션 (FANTOM FOUNDATION) [GB/GB]; 2510 그랜드 카이만 케이와이 1-1104, Grand Cayman (KY). 연세대학교 산학협력단 (INDUSTRY-ACADEMIC COOPERATION FOUNDATION, YONSEI UNIVERSITY) [KR/KR]; 03722 서울시 서대문구 연세로 50, Seoul (KR).

- (72) 발명자: 안병익 (AHN, Byung Ik); 13589 경기도 성남시 분당구 중앙공원로 54, 213동 1701호, Gyeonggi-do (KR). 한요섭 (HAN, Yo Sub); 03310 서울시 은평구 진관1로 77-8, Seoul (KR). 최상민 (CHOI, Sang Min); 03334 서울시 은평구 연서로23길 3-9, 502호, Seoul (KR). 박지호 (PARK, Ji Ho); 12771 경기도 광주시 오포읍 신현로 65-24, 205동 604호, Gyeonggi-do (KR). 장기영 (JANG, Ki Young); 10510 경기도 고양시 덕양구 토당로32번길 11, 101동 803호, Gyeonggi-do (KR). 천현준 (CHEON, Hyun Joon); 21458 인천시 부평구 동수로120번길 43, Incheon (KR).
- (74) 대리인: 특허법인 광장리앤코 (LEE & KO IP); 04532 서울시 중구 남대문로 63, 25층, 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, CZ, DE, DJ, DK, DM, DO, DZ, EC,

(54) Title: DISTRIBUTED DATA SYSTEM AND DATA DISTRIBUTION STORAGE METHOD

(54) 발명의 명칭: 분산 데이터 시스템 및 데이터 분산 저장 방법



(57) Abstract: The data distribution storage method in a first node according to the present disclosure comprises the steps of: determining at least one reference event block for generating a first event block; determining time information of the first event block on the basis of time information corresponding to the at least one reference event block; generating the first event block on the basis of the determined time information; and updating a flag table including information on an event block reachable from the first event block, among event blocks determined as a root of a current frame, on the basis of information corresponding to the generated first event block.

EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, 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, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, 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, 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, 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))

(57) 요약서: 본 개시에 따른 제1 노드에서의 데이터 분산 저장 방법은 제1 이벤트 블록 생성을 위한 적어도 하나의 참조 이벤트 블록을 결정하는 단계, 적어도 하나의 참조 이벤트 블록에 대응하는 시간 정보에 기초하여 제1 이벤트 블록의 시간 정보를 결정하는 단계, 결정된 시간 정보를 기반으로 제1 이벤트 블록을 생성하는 단계 및 생성된 제1 이벤트 블록에 대응하는 정보를 기반으로 현재 프레임의 루트로 결정된 이벤트 블록 중 제1 이벤트 블록에서 도달 가능한 이벤트 블록의 정보를 포함하는 플래그 테이블을 업데이트하는 단계를 포함한다.

명세서

발명의 명칭: 분산 데이터 시스템 및 데이터 분산 저장 방법

기술분야

- [1] 본 개시는 분산 데이터 시스템 및 데이터 분산 저장을 위한 방법 및 장치에 관한 것이다.

배경기술

- [2] 종래의 중앙화된 데이터 시스템은 서버가 네트워크로 연결된 노드들에 관한 데이터를 수집하여 저장하는 방식의 시스템으로, 데이터의 관리는 용이하였으나 서버에 보안 상 문제가 생기는 경우 시스템 전체의 데이터가 손실될 우려가 있었다.
- [3] 이러한 문제를 해결하고자 분산 데이터 시스템이 개발되었다. 분산 데이터 시스템은 P2P(peer-to-peer) 네트워크로 연결된 모든 노드가 데이터를 발생시키는 주체인 동시에 동일한 데이터 기록을 가지는 시스템을 의미한다. 분산 데이터 시스템은 데이터 기록을 순차적으로 저장하는 일종의 분산 장부(분산 데이터베이스)를 관리하며, 가상화폐, IoT, 물류관리, 고객관리, 금융 시스템 등 다양한 분야에서 새로운 솔루션을 제공할 수 있는 시스템으로 관심 받고 있다.
- [4] 한편 분산 데이터 시스템은 특정 노드가 잘못된 값을 전달하여 전체 분산 데이터 시스템에 잘못된 값이 공유될 가능성이 존재하기 때문에, 종래의 중앙화된 데이터 시스템과 비교하여 데이터의 신뢰성이 문제가 될 수 있다. 따라서, 분산 데이터 시스템은 신뢰성을 보장하기 위해 pBFT(Practical Byzantine Fault Tolerance) 합의 알고리즘이 적용될 수 있다. pBFT 합의 알고리즘이란 네트워크 내에서 특정 노드가 잘못된 값을 다른 노드들에게 전달하여도 전체 분산 데이터 시스템에는 잘못된 값으로 인한 영향 없이 정확한 값이 합의에 이른다는 것을 보장하는 알고리즘을 의미한다. 다시 말해, pBFT 합의 알고리즘은 합의를 이루는 노드 사이에 일부의 노드가 신뢰할 수 없는 경우에도 분산 데이터 시스템의 신뢰성에는 문제가 없도록 합의를 이뤄낼 수 있는 알고리즘이다.
- [5] pBFT 합의 알고리즘은 참여하는 노드가 제한된 허가형(permissioned) 분산 데이터 시스템을 위한 합의 알고리즘이며, 비동기 네트워크(Asynchronous Network) 환경을 전제로 한다. pBFT 합의 알고리즘으로는 대표적으로 Tangle, Hashgraph 등이 있으나, 합의에 도달하기 위해 많은 시간이 소요되는 점 및 각 노드가 합의를 도출하기 위해 복잡한 연산을 수행해야 한다는 점 등을 극복하기 위한 새로운 알고리즘에 대한 연구가 진행되고 있다.

발명의 상세한 설명

기술적 과제

- [6] 분산 데이터 시스템 및 데이터 분산 저장 방법 및 이를 이용한 장치를 제공하는

데 있다. 본 실시예가 이루고자 하는 기술적 과제는 상기된 바와 같은 기술적 과제들로 한정되지 않으며, 이하의 실시예들로부터 또 다른 기술적 과제들이 유추될 수 있다.

과제 해결 수단

- [7] 일 측면에 따른, 본 개시에 따른 제1 노드에서의 데이터 분산 저장 방법은 제1 이벤트 블록 생성을 위한 적어도 하나의 참조 이벤트 블록을 결정하는 단계, 적어도 하나의 참조 이벤트 블록에 대응하는 시간 정보에 기초하여 제1 이벤트 블록의 시간 정보를 결정하는 단계, 결정된 시간 정보를 기반으로 제1 이벤트 블록을 생성하는 단계 및 생성된 제1 이벤트 블록에 대응하는 정보를 기반으로 현재 프레임의 루트로 결정된 이벤트 블록 중 제1 이벤트 블록에서 도달 가능한 이벤트 블록의 정보를 포함하는 플래그 테이블을 업데이트하는 단계를 포함할 수 있다.
- [8] 다른 일 측면에 따른 2 개 이상의 노드가 포함된 분산 데이터 시스템에서 노드는 다른 노드와 데이터를 송수신하는 통신부, 인트럭션을 저장하는 메모리 및 인스트럭션을 처리하는 프로세서를 포함하고, 프로세서는 일 측면에 따른 데이터 분산 저장 방법의 단계들을 수행할 수 있다.
- [9] 또 다른 측면에 따르면, 상기 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공한다.

발명의 효과

- [10] 상기된 바에 따르면, 분산 데이터 시스템은 합의된 데이터의 신뢰성을 보장하면서, 분산 데이터 시스템에 포함된 노드들이 합의를 도출하는 데 소요되는 시간과 노드들의 연산량을 줄이는 효과를 얻을 수 있다.
- [11] 또한, 데이터 분산 저장 방법은 불필요한 데이터를 삭제하는 방법을 수행할 수 있으므로, 노드에 저장되는 데이터의 양을 줄이는 효과를 얻을 수 있다.

도면의 간단한 설명

- [12] 도 1은 일 실시예에 따른 이벤트 블록 체인을 설명하기 위한 도면이다.
- [13] 도 2는 일 실시예에 따른 이벤트 블록의 데이터 구조를 설명하기 위한 도면이다.
- [14] 도 3은 일 실시예에 따른 램포트 타임스탬프 알고리즘을 설명하기 위한 도면이다.
- [15] 도 4는 다른 일 실시예에 따른 이벤트 블록 체인을 설명하기 위한 도면이다.
- [16] 도 5는 일 실시예에 따른 계층적 합의 순서를 설명하기 위한 도면이다.
- [17] 도 6은 일 실시예에 따라 노드에 저장된 데이터를 설명하기 위한 도면이다.
- [18] 도 7은 일 실시예에 따라 이벤트 블록을 생성할 때 각 노드가 데이터를 저장하는 방법을 설명하기 위한 도면이다.
- [19] 도 8은 일 실시예에 따른 노드가 플래그 테이블을 이용하여 루트를 결정하는 방법을 설명하기 위한 도면이다.

- [20] 도 9는 일 실시예에 따른 노드가 클로토 체크 목록을 이용하여 클로토를 결정하는 방법을 설명하기 위한 도면이다.
- [21] 도 10은 일 실시예에 따른 노드가 새로운 프레임 항목에 루트를 저장하는 방법을 설명하기 위한 도면이다.
- [22] 도 11은 일 실시예에 따른 노드가 클로토를 결정하는 방법을 설명하기 위한 도면이다.
- [23] 도 12a 내지 도 12c는 일 실시예에 따른 노드가 최소 시간 값을 결정하는 방법을 설명하기 위한 도면이다.
- [24] 도 13a 내지 도 13c는 일 실시예에 따른 노드가 합의 시간 값을 결정하는 방법을 설명하기 위한 도면이다.
- [25] 도 14a 및 도 14b는 일 실시예에 따른 노드가 최소 시간 값의 재선택을 수행하는 방법을 설명하기 위한 도면이다.
- [26] 도 15는 일 실시예에 따른 합의에 도달한 이벤트 블록들 간의 순서를 결정하는 방법을 설명하기 위한 도면이다.
- [27] 도 16은 일 실시예에 따른 노드가 플래그 테이블을 이용하여 참조 이벤트 블록을 결정하는 방법을 설명하기 위한 도면이다.
- [28] 도 17은 일 실시예에 따른 노드가 높이 벡터 및 진입 차수 벡터를 이용하여 참조 이벤트 블록을 결정하는 방법을 설명하기 위한 도면이다.
- [29] 도 18a 내지 도 18d는 일 실시예에 따른 노드가 다른 노드와 데이터를 송수신하는 방법을 설명하기 위한 도면이다.
- [30] 도 19a 및 도 19b는 일 실시예에 따른 합의 알고리즘을 설명하기 위한 도면이다.
- [31] 도 20은 일 실시예에 따른 데이터 분산 저장 방법을 설명하기 위한 흐름도이다.
- [32] 도 21은 일 실시예에 따른 분산 데이터 시스템에 포함된 노드의 구조를 설명하기 위한 블록도이다.

발명의 실시를 위한 최선의 형태

- [33] 일 측면에 따른, 본 개시에 따른 제1 노드에서의 데이터 분산 저장 방법은 제1 이벤트 블록 생성을 위한 적어도 하나의 참조 이벤트 블록을 결정하는 단계, 적어도 하나의 참조 이벤트 블록에 대응하는 시간 정보에 기초하여 제1 이벤트 블록의 시간 정보를 결정하는 단계, 결정된 시간 정보를 기반으로 제1 이벤트 블록을 생성하는 단계 및 생성된 제1 이벤트 블록에 대응하는 정보를 기반으로 현재 프레임의 루트로 결정된 이벤트 블록 중 제1 이벤트 블록에서 도달 가능한 이벤트 블록의 정보를 포함하는 플래그 테이블을 업데이트하는 단계를 포함할 수 있다.

발명의 실시를 위한 형태

- [34] 본 실시예들에서 사용되는 용어는 본 실시예들에서의 기능을 고려하면서 가능한 현재 널리 사용되는 일반적인 용어들을 선택하였으나, 이는 당 기술분야에 종사하는 기술자의 의도 또는 판례, 새로운 기술의 출현 등에 따라

달라질 수 있다. 또한, 특정한 경우는 임의로 선정된 용어도 있으며, 이 경우 해당 실시예의 설명 부분에서 상세히 그 의미를 기재할 것이다. 따라서, 본 실시예들에서 사용되는 용어는 단순한 용어의 명칭이 아닌, 그 용어가 가지는 의미와 본 실시예들의 전반에 걸친 내용을 토대로 정의되어야 한다.

[35] 실시예들에 대한 설명들에서, 어떤 부분이 다른 부분과 연결되어 있다고 할 때, 이는 직접적으로 연결되어 있는 경우뿐 아니라, 그 중간에 다른 구성요소를 사이에 두고 전기적으로 연결되어 있는 경우도 포함한다. 또한 어떤 부분이 어떤 구성요소를 포함한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다. 또한, 실시예들에 기재된 "...부", "...모듈"의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어 또는 소프트웨어로 구현되거나 하드웨어와 소프트웨어의 결합으로 구현될 수 있다. 뿐만 아니라, 구성요소들 및 '~부'들은 디바이스 또는 보안 멀티미디어카드 내의 하나 또는 그 이상의 CPU들을 재생시키도록 구현될 수도 있다.

[36] 본 실시예들에서 사용되는 "구성된다" 또는 "포함한다" 등의 용어는 명세서 상에 기재된 여러 구성 요소들, 또는 여러 단계들을 반드시 모두 포함하는 것으로 해석되지 않아야 하며, 그 중 일부 구성 요소들 또는 일부 단계들은 포함되지 않을 수도 있고, 또는 추가적인 구성 요소 또는 단계들을 더 포함할 수 있는 것으로 해석되어야 한다.

[37] 또한 실시 예를 설명하기 위한 순서도의 각 단계와 이 단계의 조합들은 컴퓨터 프로그램 인스트럭션들에 의해 수행될 수 있음을 이해할 수 있을 것이다. 이들 컴퓨터 프로그램 인스트럭션들은 범용 컴퓨터, 특수용 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비의 프로세서에 탑재될 수 있으므로, 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비의 프로세서를 통해 수행되는 그 인스트럭션들이 흐름도 블록(들)에서 설명된 기능들을 수행하는 수단을 생성하게 된다. 이들 컴퓨터 프로그램 인스트럭션들은 특정 방식으로 기능을 구현하기 위해 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비를 지향할 수 있는 컴퓨터 이용 가능 또는 컴퓨터 판독 가능 메모리에 저장되는 것도 가능하므로, 그 컴퓨터 이용 가능 또는 컴퓨터 판독 가능 메모리에 저장된 인스트럭션들은 흐름도 블록(들)에서 설명된 기능을 수행하는 인스트럭션 수단을 내포하는 제조 품목을 생산하는 것도 가능하다. 컴퓨터 프로그램 인스트럭션들은 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비 상에 탑재되는 것도 가능하므로, 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비 상에서 일련의 동작 단계들이 수행되어 컴퓨터로 실행되는 프로세스를 생성해서 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비를 수행하는 인스트럭션들은 흐름도 블록(들)에서 설명된 기능들을 실행하기 위한 단계들을 제공하는 것도 가능하다.

[38] 또한, 각 블록은 특정된 논리적 기능(들)을 실행하기 위한 하나 이상의 실행

가능한 인스트럭션들을 포함하는 모듈, 세그먼트 또는 코드의 일부를 나타낼 수 있다. 또, 몇 가지 대체 실행 예들에서는 블록들에서 언급된 기능들이 순서를 벗어나서 발생하는 것도 가능함을 주목해야 한다. 예컨대, 잇달아 도시되어 있는 두 개의 블록들은 사실 실질적으로 동시에 수행되는 것도 가능하고 또는 그 블록들이 때때로 해당하는 기능에 따라 역순으로 수행되는 것도 가능하다.

- [39] 하기 실시예들에 대한 설명은 권리범위를 제한하는 것으로 해석되지 말아야 하며, 해당 기술분야의 당업자가 용이하게 유추할 수 있는 것은 실시예들의 권리범위에 속하는 것으로 해석되어야 할 것이다. 이하 첨부된 도면들을 참조하면서 오로지 예시를 위한 실시예들을 상세히 설명하기로 한다.
- [40] 도 1은 일 실시예에 따른 이벤트 블록 체인을 설명하기 위한 도면이다.
- [41] 도 1은 A 노드 내지 D 노드가 생성한 복수의 이벤트들로 구성된 이벤트 블록 체인을 나타낸다. 여기서 노드는 사용자에게 의해 이벤트 블록을 생성하고, 다른 노드들과 통신하여 이벤트 블록 간의 합의 시간 값을 도출하는 장치로서, 실시예에서 노드는 스마트폰, 태블릿 PC, 휴대폰, PDA(personal digital assistant), 랩톱 및 기타 모바일 컴퓨팅 장치 중 적어도 하나를 포함하는 연산 장치이며, 위에서 언급한 장치 이외에도 연산을 수행하고 다른 노드와 통신을 수행할 수 있는 전자 장치를 포함할 수 있다.
- [42] 도 1에서 원은 하나의 이벤트 블록으로, 상기 이벤트 블록은 복수의 거래 정보를 포함할 수 있으며, 합의 알고리즘을 통해 합의 시간 값이 결정되는 데이터의 단위를 의미한다. 도 1의 각 원에 작성된 영문 소문자는 해당 이벤트 블록을 생성한 노드를 나타내며, 숫자는 해당 노드가 생성한 이벤트 블록이 생성된 순서를 기반으로 부여되며, 이를 기반으로 해당 노드가 생성한 이벤트 블록의 수를 알 수 있다. 예를 들어, 도 1에서 'c1'으로 표시된 원은 C 노드가 첫 번째로 생성한 이벤트 블록을 의미한다.
- [43] 또한 각 노드는 이벤트 블록을 생성할 때, 자신 혹은 다른 노드가 이전에 생성된 이벤트 블록 중 적어도 하나를 참조할 수 있다. 일 예로 각 노드는 자신이 생성한 이벤트 블록들 중에서 가장 최신의 이벤트 블록을 포함하여 둘 이상의 이벤트 블록을 참조할 수 있다. 여기서, 참조되는 이벤트 블록의 수는 시스템에 따라 변경될 수 있다. 또한 참조되는 이벤트 블록의 수는 고정될 수 있고, 각 이벤트 블록마다 달라질 수 있다. 그러나 본 개시의 실시예에서는 설명의 편의를 위해, 참조하는 이벤트 블록의 수는 고정된 것으로 가정하나, 실시예에 따라서 노드는 새로운 이벤트 블록을 생성할 때 참조할 이벤트 블록의 개수를 판단하는 절차를 추가로 수행할 수도 있다.
- [44] 도 1을 참고하면, 참조되는 이벤트 블록들은 화살표로 연결될 수 있다. 예를 들어, C 노드는 세 번째 타임 슬롯에 c2를 생성할 수 있고, 이 때 참조하는 이벤트 블록의 개수를 3개라고 하면, 이때 c2의 참조 이벤트 블록은 C 노드 자신이 가장 최근에 생성한 이벤트 블록(c1)과 A 노드와 D 노드가 생성한 이벤트 블록 중 가장 최신의 이벤트 블록(a2 및 d1)일 수 있다. 이때 C 노드는 c2의 참조 이벤트

블록으로서 자신이 가장 최근에 생성한 이벤트 블록 외에 다른 노드들에 의해 생성된 이벤트 블록들을 선택할 수 있는데, 여기서 참조 이벤트 블록은 본 개시의 다양한 실시예를 통해 선택될 수 있다.

- [45] 한편 도 1과 같은 이벤트 블록 체인에서 가장 오른쪽에 표시되는 이벤트 블록은 가장 최근에 생성된 이벤트 블록을 의미하고, 가장 왼쪽에 표시되는 이벤트 블록은 가장 먼저 생성된 이벤트 블록을 의미한다. 도 1에 도시된 바와 같이, 일 실시예에 따른 각 노드들은 비동기적으로 이벤트 블록들을 생성하며, 이벤트 블록 체인은 DAG(Directed acyclic graph, 방향성 비순환 그래프) 기반의 그래프로 표시될 수 있다.
- [46] 도 2는 일 실시예에 따른 이벤트 블록의 데이터 구조를 설명하기 위한 도면이다.
- [47] 도 2를 참조하면, 일 실시예에 따른 이벤트 블록은 합의 시간 값이 결정되는 데이터의 단위로서, 서명 정보, 타임스탬프, 거래 정보 및 해시 벡터 중 적어도 하나를 포함할 수 있다.
- [48] 실시 예에서 서명 정보는 해당 이벤트 블록을 생성한 노드에 관한 정보를 포함하며, 하나의 노드가 생성한 서로 다른 이벤트 블록들은 동일한 서명 정보를 포함할 수 있다.
- [49] 타임스탬프는 이벤트 블록이 생성된 시점을 나타내는 정보를 포함한다. 상기 타임스탬프를 기반으로 합의 시간 값을 도출할 수 있다. 실시 예에서 타임스탬프는 논리적 시간 값 및 물리적 시간 값 중 적어도 하나를 포함할 수 있다. 논리적 시간 값은 이벤트 블록 체인에 포함된 일부 이벤트 블록들 간의 발생 전후 관계(happened-before relation)에 기초하여, 부분적인 순서를 결정하기 위해 활용되는 시간 정보를 의미한다. 본 개시의 일 실시예는 논리적 시간 값을 획득하기 위해, 램포트 타임스탬프(Lamport timestamp) 알고리즘을 이용한다. 일 실시 예에 따르면 새로운 이벤트 블록을 생성할 때 상기 생성되는 이벤트 블록의 참조 이벤트 블록의 논리적 시간 값들 중 가장 큰 값 보다 하나 증가된 값이 논리적 시간 값에 대응할 수 있다.
- [50] 한편, 물리적 시간 값은 이벤트 블록의 생성 당시 노드의 실제 시간 정보를 의미한다. 이와 같은 시간 물리적 시간 값은 시간 값 그대로 저장되거나 해시 값 형태로 저장될 수 있다.
- [51] 거래 정보(transaction information)는 각 노드에 의해 수행된 실제 거래 이력 정보를 포함하며, 거래 정보의 크기는 가변적이나 노드 및 분산 데이터 시스템의 성능에 따라 제한될 수 있다.
- [52] 해시 벡터(hash vector)는 해당 이벤트 블록의 참조 이벤트 블록들에 관한 해시 정보를 포함한다. 여기서 참조 이벤트 블록의 수는 달라질 수 있으므로, 해시 벡터의 크기는 가변적일 수 있다.
- [53] 도 3은 일 실시예에 따른 램포트 타임스탬프 알고리즘을 설명하기 위한 도면이다.

- [54] 도 3을 참조하면, 노드 A 내지 노드 E가 생성한 이벤트 블록이 개시된다. 본 개시의 일 실시예에 따른 이벤트 블록은 논리적 시간 값으로서 램포트 타임스탬프 값을 활용할 수 있으며, 각 이벤트 블록의 램포트 타임스탬프 값은 아래의 램포트 타임스탬프 알고리즘에 따라 결정될 수 있다.
- [55] 먼저, 각 노드가 첫 번째로 생성한 이벤트 블록의 램포트 타임스탬프 값은 1로 결정된다. 이후, 각 이벤트 블록의 램포트 타임스탬프 값은 참조 이벤트 블록의 램포트 타임스탬프 값에 따라 결정될 수 있다. 구체적으로, 이벤트 블록의 램포트 타임스탬프 값은 참조 이벤트 블록들의 램포트 타임스탬프 값 중 최대값보다 1이 큰 값으로 결정된다. 실시 예에 따라 각 노드가 첫 번째로 생성한 이벤트 블록의 타임스탬프 값은 0 부터 시작할 수도 있다.
- [56] 각 원은 이벤트 블록을 나타내고, 원 안의 영문 소문자는 해당 이벤트 블록을 생성한 노드를 나타내며, 숫자는 해당 노드가 생성한 이벤트 블록이 생성된 순서를 의미한다. 또한, 괄호 안에 기재된 숫자는 램포트 타임스탬프 값을 나타낸다. 또한, 동일한 노드에 의해 생성된 이벤트 블록은 같은 열에 표시될 수 있다.
- [57] 일 예에서, C 노드가 가장 최근에 생성한 이벤트 블록(310)의 램포트 타임스탬프 값은 해당 이벤트 블록(310)이 참조하는 이벤트 블록들(320)의 램포트 타임스탬프 값 4, 2, 3 중 최대값인 4 보다 1이 큰 5로 결정될 수 있다. 이러한 방식으로 결정된 이벤트 블록의 램포트 타임스탬프 값은 이벤트 블록이 생성된 시점의 선후 관계를 결정하는 데 활용될 수 있다.
- [58] 한편, 도 3은 C 노드가 가장 최근에 생성한 이벤트 블록(310)에 저장된 데이터(330)를 함께 나타낸다. C 노드가 가장 최근에 생성한 이벤트 블록(310)에는 C 노드의 서명 정보, 램포트 타임스탬프 값(5), 거래 정보 및 참조 이벤트 블록들(320)의 해시 벡터가 저장될 수 있다. 이와 같이 각 이벤트 블록은 해당 이벤트 블록에 대응하는 데이터 값을 가질 수 있으며, 각 노드는 해당 노드가 생성한 이벤트 블록에 대응하는 데이터 값을 저장할 수 있다.
- [59] 도 4는 다른 일 실시예에 따른 이벤트 블록 체인을 설명하기 위한 도면이다.
- [60] 도 4를 참조하면 총 5개의 노드(A 노드 내지 E 노드)로 이루어진 분산 데이터 시스템의 이벤트 블록 체인의 일 예시를 나타낸다. A 노드 내지 E 노드가 처음 생성한 각각의 이벤트 블록은 기초 이벤트 블록(leaf event block)(410)으로 정의될 수 있으며, 기초 이벤트 블록(410)의 램포트 타임스탬프 값은 1로 정의된다. 각 노드는 기초 이벤트 블록을 생성한 이후 비동기적으로 이벤트 블록들을 생성할 수 있다. 예를 들어, C 노드는 자신이 생성한 기초 이벤트 블록(c1), B 노드의 기초 이벤트 블록(b1) 및 D 노드의 기초 이벤트 블록(d1)을 참조하여 두 번째 이벤트 블록(c2)을 생성할 수 있다. 이때, c2의 램포트 타임스탬프 값은 램포트 타임스탬프 알고리즘에 의해 결정되며, c2가 참조하는 이벤트 블록의 타임스탬프 값이 모두 1인바, c2의 램포트 타임스탬프 값은 2가 된다.
- [61] 합의 시간 값을 도출하기 위해, 이벤트 블록들은 일정 조건에 따라 분류될 수

있다. 일 실시예에 따르는 노드는 이벤트 블록 체인에 포함된 이벤트 블록들 중에서 일부 이벤트 블록들을 '루트(root)'로 결정할 수 있다. 루트는 (1) 각 노드들의 기초 이벤트 블록이거나 (2) 이벤트 블록이 속한 프레임에 포함된 루트들 중 전체 노드의 개수의 2/3 보다 많은 루트들에 도달할 수 있는(reachable) 이벤트 블록들을 의미한다. 또한 이와 같이 결정된 루트에 해당하는 이벤트 블록은 새로운 프레임에 포함될 수 있다. 여기서 'b 이벤트 블록이 a 이벤트 블록에 도달할 수 있다'는 것은 b 이벤트 블록이 a 이벤트 블록을 직간접적으로 참조하여 생성되었다는 것을 의미한다. 이를 도식화하면, b 이벤트 블록과 a 이벤트 블록이 직접 연결되어 있거나, b 이벤트 블록이 여러 이벤트 블록들을 통해서 a 이벤트 블록이 연결된 것으로 표현할 수 있다. 도 4에서 d2는 c2에 도달 가능한 이벤트 블록일 뿐 아니라, e1에 도달 가능한 이벤트 블록이다. 또한 d2는 c2가 참조하는 b1, c1 및 d1에도 도달할 수 있는 이벤트 블록이다. 일 실시예에 따른 노드는 이벤트 블록을 생성한 후, 플래그 테이블을 이용하여 해당 이벤트 블록이 루트인지 판단할 수 있다. 실시 예에서 각 노드는 획득한 정보를 기반으로 다른 노드가 생성한 이벤트 블록 역시 특정 프레임 내에서 루트인지 여부를 판단할 수 있다.

[62] 일 실시예에 따른 데이터 분산 저장 방법에서 루트들을 프레임으로 분류될 수 있다. 각 노드의 기초 이벤트 블록들은 루트로 결정되며, 각 노드들의 첫 번째 루트들은 제1 프레임(f1)에 속한 루트들로 결정될 수 있다. 이후 각 노드들에 의해 두 번째로 형성되는 루트들은 제2 프레임(f2)에 속한 루트들이 될 수 있다. 또한, 제2 프레임(f2)에 속한 루트들과 제1 프레임(f1)에 속한 루트들 사이에 포함되는 이벤트 블록들은 제1 프레임(f1)에 속한 이벤트 블록일 수 있다. 도 4를 참고하면, a2와 c2는 제1 프레임(f1)에 속한 이벤트 블록들이다.

[63] 또한 일 실시예에 따른 각 노드는 루트들 중에서 일정 조건을 만족하는 일부 루트를 '클로토(Clotho)'로 결정할 수 있다. 다시 말해, 루트로 결정된 이벤트 블록들 중 일부가 클로토로 결정될 수 있다. 노드는 이전에 생성되었던 이벤트 블록의 정보가 전체 노드 중 2/3 보다 많은 노드들에게 공유되었다는 것이 확실하게 보장될 때, 해당 이벤트 블록을 클로토로 결정할 수 있다. 일 실시예에 따른 노드는 각 노드에 저장된 클로토 체크 목록을 이용하여 클로토를 결정할 수 있다. 각 노드는 클로토로 결정된 이벤트 블록들에 대하여 최소 시간 값을 계산할 수 있다. 클로토의 최소 시간 값은 합의 알고리즘의 최종 결과 값인 '아트로포스(Atropos)의 합의 시간 값'을 결정하는 데 사용될 수 있다. 실시 예에서 클로토 및 아트로포스와 같은 용어는 해당 특징을 설명하기 위한 용어로 다른 용어로 대체될 수 있으며, 대응되는 이벤트 블록은 다른 이벤트 블록과의 관계 안에서 결정될 수 있다.

[64] 한편 일 실시예에 따른 각 노드는 클로토들 중에서 합의가 완료된 클로토를 '아트로포스'로 결정할 수 있다. 다시 말해, 클로토로 결정된 이벤트 블록들 중 일부가 아트로포스로 결정될 수 있다. 이때 아트로포스로 구성된 이벤트 블록

체인은 메인 체인으로 정의될 수 있다. 또한, 아트로포스의 참조 이벤트 블록들은 동일한 '아트로포스의 합의 시간 값'을 갖게 되며, '아트로포스의 합의 시간 값'을 이용하여 각 이벤트 블록들 간의 순서를 결정하게 된다.

- [65] 즉, 일 실시예에 따른 데이터 분산 저장 방법은 계층적 합의 순서에 따라 각 이벤트 블록이 생성된 순서를 결정하기 위해, 이벤트 블록을 루트, 클로토 및 아트로포스로 분류하고, 클로토의 최소 시간 값 및 아트로포스의 합의 시간 값을 정의할 수 있다. 계층적 합의 순서에 관해서는 도 5 및 관련 단락에서 설명하기로 한다.
- [66] 도 5는 일 실시예에 따른 계층적 합의 순서를 설명하기 위한 도면이다.
- [67] 도 5를 참조하면, 일 실시예에 따른 데이터 분산 저장 방법은 3 단계로 이루어진 계층적 합의 순서(topological consensus ordering)에 따라 복수의 이벤트 블록들 간의 순서를 결정할 수 있다. 실시 예에서 합의 순서를 결정하는 절차는 각 노드에서 수행될 수 있다.
- [68] S510에서 각 이벤트 블록이 가진 아트로포스의 합의 시간 값이 동일한지 판단할 수 있다. 판단 결과, 각 이벤트 블록의 아트로포스의 합의 시간 값이 다르다면, S515에서 아트로포스의 합의 시간 값이 이른 이벤트 블록이 더 먼저 생성된 이벤트 블록인 것으로 결정할 수 있다.
- [69] 만약 각 이벤트 블록이 가진 아트로포스의 합의 시간 값이 동일한 경우, S520에서 각 이벤트 블록의 램포트 타임스탬프 값이 동일한지 판단할 수 있다. 판단 결과, 각 이벤트 블록의 램포트 타임스탬프 값이 다르다면, S525에서 램포트 타임스탬프 값이 작은 이벤트 블록이 먼저 생성된 이벤트 블록인 것으로 결정할 수 있다.
- [70] 만약, 각 이벤트 블록이 가진 아트로포스의 합의 시간 값과 램포트 타임스탬프 값이 동일한 경우, S530에서 각 이벤트 블록의 해시 값을 비교하여, 해시 값이 이른 이벤트 블록이 더 먼저 생성된 이벤트 블록인 것으로 결정할 수 있다. 여기서 해시 값은 이벤트 블록의 물리적 시간 정보 값 및 거래 정보 중 적어도 하나에 기초하여 생성될 수 있다.
- [71] 도 6은 일 실시예에 따라 노드에 저장된 데이터를 설명하기 위한 도면이다.
- [72] 도 6을 참조하면, 일 실시예에 따른 노드는 이벤트 블록을 생성하고 다른 노드들에 의해 생성된 이벤트 블록들과 자신이 생성한 이벤트 블록들의 생성된 시점의 순서에 관하여 합의를 도출하기 위해 특정 데이터들을 저장할 수 있다. 노드는 각 노드에 대응하는 높이 벡터, 각 노드에 대입하는 진입 차수 벡터, 프레임 정보, 클로토 체크 목록, 메인 체인 정보, 최소 시간 정보, 이벤트 블록 체인 정보 및 플래그 테이블 중 적어도 하나를 저장할 수 있으나, 노드에 저장되는 데이터는 이에 제한되지 않는다. 상기 데이터 중 일부 데이터는 실시 예에 따라 노드에 저장되지 않을 수도 있다. 또한 실시 예에서 새로운 이벤트 블록을 생성하기 위해서 참조되는 다른 이벤트 블록에 대응하는 노드와 통신할 때 상기 저장된 정보 중 적어도 일부를 상기 통신하는 노드에 전달할 수도 있다.

- [73] 프레임 정보는 현재 노드가 알고 있는 각 프레임에 속한 루트들의 정보를 포함할 수 있다. 보다 구체적으로 실시 예에서 A 노드가 알고 있는 각 프레임의 루트 이벤트 블록이 프레임 정보에 포함될 수 있다.
- [74] 클로토 체크 목록은 각 프레임에 속한 루트들이 클로토 후보(CC; Clotho Candidate) 및 클로토에 해당되는지 여부를 나타내는 목록으로, 클로토 후보 플래그 및 클로토 플래그를 포함한다. 일 실시예에 따른 노드는 새로운 이벤트 블록을 생성하고 난 후, 클로토 체크 목록을 통해 클로토를 결정할 수 있다.
- [75] 메인 체인 정보는 현재 노드가 알고 있는 아트로포스들로 이루어진 이벤트 블록 체인을 의미한다. 메인 체인 정보는 아트로포스의 해시 벡터와 합의 시간 값을 대응시켜 저장할 수 있다.
- [76] 이벤트 블록 체인 정보는 현재 노드가 알고 있는 이벤트 블록들로 이루어진 이벤트 블록 체인을 의미하며, 그래프 형태로 저장될 수 있다. 예를 들어, 그래프 형태는 DAG 기반의 그래프 형태일 수 있으나, 이에 제한되지 않는다.
- [77] 플래그 테이블은 현재 프레임의 루트들 중 도달 가능한 루트들을 각 노드 별로 저장한 테이블을 의미한다. 실시 예에서 플래그 테이블은 A노드가 알고 있는 각 노드가 도달 가능한 루트들의 집합에 관한 정보를 포함할 수 있다. 플래그 테이블은 이벤트 블록을 생성하기 위해 다른 노드와 정보를 교환하는 과정에서 수신된 정보를 기반으로 업데이트 될 수 있다.
- [78] 일 실시예에 따른 각 노드는 플래그 테이블을 이용하여 클로토 체크 목록을 업데이트할 수 있으며, 이에 대한 구체적인 방법은 도 9 및 관련 단락에서 설명하기로 한다. 또한, 각 노드는 이벤트 블록을 생성할 때, 플래그 테이블을 이용하여 참조 이벤트 블록을 결정할 수 있으며, 이에 대한 구체적인 방법은 도 16 및 관련 단락에서 설명하기로 한다.
- [79] 최소 시간 정보는 이벤트 블록이 클로토로 결정되고 난 후, 각 클로토에 대응하는 계산되는 시간 정보 값을 의미하며 합의 알고리즘의 최종 결과 값인 '아트로포스의 합의 시간 값'을 결정하는 데 사용되는 값이다. 최소 시간 값을 구하는 구체적인 방법은 도 12a 내지 도 12c 및 관련 단락에서 설명하기로 한다.
- [80] 이외에도 각 노드는 높이 벡터 및 진입 차수 벡터를 더 저장할 수 있다. 여기서 높이 벡터는 각 노드가 생성한 이벤트 블록의 수를 의미하며, 진입 차수 벡터는 각 노드가 생성한 가장 최근의 이벤트 블록들을 참조하는 이벤트 블록의 수를 의미한다. 높이 벡터 및 진입 차수 벡터의 크기는 합의 알고리즘에 참여하는 노드의 수에 대응될 수 있다. 다른 일 실시예에 따르는 합의 알고리즘에 참여하는 각 노드는 이벤트 블록을 생성할 때, 높이 벡터 및 진입 차수 벡터 값 중 적어도 하나에 기초하여, 참조 이벤트 블록을 결정할 수 있다. 높이 벡터 및 진입 차수 벡터 값을 이용하여 참조 이벤트 블록을 결정하는 방법은 도 17 및 관련 단락에서 설명하기로 한다.
- [81] 도 7은 일 실시예에 따른 노드가 데이터를 저장하는 방법을 설명하기 위한 도면이다.

- [82] 도 7을 참조하면, A 노드 내지 E 노드가 모두 기초 이벤트 블록을 생성하고, A 노드 및 C 노드가 각각 두 번째 이벤트 블록(a2, c2)을 생성하였다고 가정했을 때, 각 이벤트 블록의 참조 관계, A 노드에 저장된 데이터(710) 및 C 노드에 저장된 데이터(720)가 도시된다.
- [83] A 노드가 가장 최근에 생성한 이벤트 블록인 a2는 a1, b1, c1를 참조하고 있으며, A 노드의 이벤트 블록 체인에는 해당 내용이 저장될 수 있다. 또한 A 노드의 플래그 테이블에는 a2의 생성 시점에 A 노드 내지 E 노드가 가장 최근에 생성한 이벤트 블록들에 대하여, 도달 가능한 루트의 집합이 저장될 수 있으므로, 플래그 테이블의 A 노드에 대응하는 항목에는 {a1, b1, c1}가 저장될 수 있다. 한편, 기초 이벤트 블록은 모든 노드들이 알고 있으므로, A 노드는 D 노드 및 E 노드가 생성한 이벤트 블록을 참조하지 않았어도, D 노드 및 E 노드의 이벤트 블록으로 각각 {d1}, {e1}이 존재함을 알 수 있다. 또한, A 노드는 프레임 정보로서 f1 프레임에 포함된 모든 루트 정보를 알 수 있다.
- [84] 또한, C 노드가 가장 최근에 생성한 이벤트 블록인 c2는 b1, c1, d1를 참조하고 있으므로, C 노드의 이벤트 블록 체인에는 해당 내용이 저장되고, 플래그 테이블의 C 노드에 대응하는 항목에는 {b1, c1, d1}가 저장될 수 있다. 그러나 C 노드는 A 노드의 이벤트 블록을 참조하지 않았으므로, A 노드가 가장 최근에 생성한 이벤트 블록은 a2가 아닌 a1인 것으로 알고 있다. 따라서, 플래그 테이블의 A 노드에 대응하는 항목에는 {a1}가 저장된다.
- [85] 도 8은 일 실시예에 따른 노드가 플래그 테이블을 이용하여 루트를 결정하는 방법을 설명하기 위한 도면이다.
- [86] 도 4에서 설명한 바와 같이, 기초 이벤트 블록은 루트로 결정될 수 있다. 그러나 그 이후 생성된 이벤트 블록이 루트로 결정되기 위해서는, 이벤트 블록이 속한 프레임에 포함된 루트들 중 2/3 보다 많은 루트들에 도달할 수 있는(reachable) 이벤트 블록이어야 한다. 일 실시예에 따른 각 노드는 이벤트 블록을 생성한 후, 플래그 테이블을 이용하여 해당 이벤트 블록이 루트인지 여부를 결정할 수 있다. 구체적으로, 분산 데이터 시스템에 포함된 노드의 수가 n 일 때, 플래그 테이블은 n 개의 항목을 구비하고, 각 항목에는 대응되는 노드에 의해 생성된 최신 이벤트 블록이 같은 프레임에 속한 루트들 중에서 도달 가능한 루트들의 집합을 저장하게 된다. 이때 플래그 테이블의 해당 노드에 대응하는 항목에 저장된 집합의 원소의 수가 $2n/3$ 개 이상이면, 해당 노드가 생성한 이벤트 블록을 루트로 결정할 수 있다.
- [87] 실시 예에서 B 노드가 a2, b1, c2를 참조하여 새로운 이벤트 블록 b2를 생성한 것을 가정할 때, B 노드에 저장된 데이터(810)를 나타낸다. B 노드에 저장된 플래그 테이블에서 B 노드에 대응하는 항목(815)에는 b2로부터 도달 가능한 루트들인 {a1, b1, c1, d1}가 저장될 수 있다. 이때 플래그 테이블의 B 노드에 대응하는 항목(815)에 저장된 루트들의 수는 총 4 개로, 분산 데이터 시스템에 포함된 노드 수(n=5)의 2/3 보다 큰 값이다($4 > 3.33$). 따라서, B 노드는 b2를

새로운 루트로 결정할 수 있다. 이러한 방식으로 각 노드는 비동기적으로 이벤트 블록을 생성하면서 각자 플래그 테이블을 이용해 자신이 생성한 이벤트 블록이 루트인지 여부를 결정할 수 있다.

[88] B 노드의 프레임 정보에는 기초 이벤트 블록들이 f1 항목에 저장될 수 있고, f1 항목에 이미 B 노드의 루트인 b1이 저장되어 있으므로, 새로운 루트 b2는 새로운 프레임 f2 항목에 저장될 수 있다.

[89] 한편, B 노드는 b2가 참조하는 a2 및 c2에 관한 정보를 업데이트하기 때문에, 이벤트 블록 체인에 관한 정보에 a2 및 c2가 참조하는 이벤트 블록에 관한 정보가 저장될 수 있다.

[90] 도 9는 일 실시예에 따른 노드가 클로토 체크 목록을 이용하여 클로토를 결정하는 방법을 설명하기 위한 도면이다.

[91] 일 실시예에 따른 노드는 이전에 생성되었던 이벤트 블록의 정보가 전체 노드 중 2/3 보다 많은 노드들에게 공유되었다는 것이 확실하게 보장될 때, 해당 이벤트 블록은 클로토로 결정될 수 있으며, 각 노드는 클로토 체크 목록을 이용하여 클로토를 결정할 수 있다. 실시 예에서 클로토 체크 목록은 각 프레임의 루트를 기준으로 업데이트 될 수 있다.

[92] 도 9는 A 노드가 a2, c2, d2를 참조하여 새로운 이벤트 블록 a3 를 생성한 것을 가정할 때, A 노드에 저장된 데이터(910)를 나타낸다. A 노드에 저장된 플래그 테이블에서 A 노드에 대응하는 항목에는 a3로부터 도달 가능한 루트들인 {a1, b1, c1, d1, e1}이 저장될 수 있다. 이때, A 노드에 대응하는 항목에 저장된 루트들의 수는 총 5 개이므로, a3는 새로운 루트로 결정될 수 있다. 또한, A 노드는 새로운 프레임 f2 항목(920)에 해당 프레임에 해당하는 루트 목록인 {a3, d2}를 저장할 수 있다. 이때, f2 항목(920)에 저장된 루트들의 수는 총 2 개로, 분산 데이터 시스템에 포함된 노드 수($n=5$)의 1/3 보다 큰 값이다($2 > 1.67$). 따라서, A 노드는 이전 프레임 f1에 포함된 루트들 중 자신이 알고 있는 루트들이 클로토 후보(CC)에 해당되는지 여부를 결정할 수 있다. 구체적으로 A 노드는 f2 항목(920)에 속한 a2와 d1 모두 도달 가능한 루트들에 대해 클로토 후보 플래그를 추가할 수 있다. 여기서 클로토 후보(CC)는 클로토 체크 목록에 포함된 정보로서, 일 실시예에 따른 클로토 후보(CC)는 플래그 테이블에 기초하여 결정될 수 있다.

[93] 구체적으로 A 노드는 A 노드에 저장된 플래그 테이블의 항목들 중에서, 전체 노드의 수(n)의 2/3 보다 많은 루트들을 저장하고 있는 항목은 선택한다. 도 9를 참고하면, 전체 노드의 수는 5 ($n=5$)이므로, 3.33 개 이상의 원소를 포함하고 있는 A 노드에 대응하는 항목{a1, b1, c1, d1, e1}과 D 노드에 대응하는 항목{b1, c1, d1, e1}이 선택된다. 이후 A 노드는 두 집합에 포함된 각 원소들 중에서 전체 노드의 수(n)의 1/3 개 이상의 항목에 포함된 원소들을 클로토 후보(CC)로 결정하고, 클로토 체크 목록에 플래그를 추가할 수 있다. 다시 말해서, 플래그 테이블의 A 노드에 대응하는 항목과 D 노드에 대응하는 항목 중 {b1, c1, d1, e1}는 각각 2

- 개의 항목에 모두 포함되어 있으므로, {b1, c1, d1, e1}를 클로포 후보로 결정하고, 클로토 체크 목록의 f1 항목에 {b1, c1, d1, e1}의 플래그를 추가할 수 있다.
- [94] 한편, A 노드는 a3가 참조하는 c2 및 d2에 관한 정보를 업데이트하기 때문에, 이벤트 블록 체인에 관한 정보에 c2 및 d2가 참조하는 이벤트 블록에 관한 정보가 저장될 수 있다.
- [95] 실시 예에서 루트를 판단하기 위한 수치 및 클로토 후보를 판단하기 위한 수치는 명세서의 설명을 위한 것으로 이는 노드 사이의 정보의 신뢰성을 확보할 수 있는 비율의 일 예시에 따라 결정된 값이다. 상기 수치들이 각 노드들이 신뢰성 있는 정보를 공유할 수 있도록 확보되는 비율에 해당할 경우 이와 같은 수치는 다소 변형되어 적용될 수 있다. 일 예로 현재 2/3 및 1/3의 수치는 비잔틴 장군 문제를 해결하기 위해 일 예시로 제안된 숫자이며, 이와 같은 숫자는 일부 변형을 통해 조절될 수 있음이 자명하다. 또한 실시 예 전반에서 루트를 판단하기 위한 기준 및 클로토 후보를 판단하기 위한 기준은 각각 이벤트 블록을 생성한 노드에서 확인한 전체 노드의 개수를 기반으로 결정되는 제1기준 및 제2기준으로 언급될 수도 있다.
- [96] 도 10은 일 실시예에 따른 노드가 새로운 프레임 항목에 루트를 저장하는 방법을 설명하기 위한 도면이다.
- [97] 도 10은 B 노드가 a3, b2, c3를 참조하여 새로운 이벤트 블록 b3를 생성한 것을 가정할 때, B 노드에 저장된 데이터(1010)를 나타낸다. B 노드에 저장된 플래그 테이블에서 B 노드에 대응하는 항목에는 프레임 f2에 해당되면서, b3로부터 도달 가능한 루트들인 {a3, b2, c3, d2}가 저장될 수 있다. 이때 플래그 테이블의 B 노드에 대응하는 항목에 저장된 루트들의 수는 총 4 개로, 분산 데이터 시스템에 포함된 노드 수($n=5$)의 2/3 보다 큰 값이다($4 > 3.33$). 따라서, B 노드는 b3를 새로운 루트로 결정할 수 있다.
- [98] 또한 B 노드의 프레임 정보 중 f2 항목에는 이미 B 노드의 루트인 b2이 저장되어 있으므로, 새로운 루트 b3은 새로운 프레임 f3 항목에 저장될 수 있다.
- [99] 한편 B 노드는 f2에 속한 {a3, b2, c3, d2}들을 생성한 노드인 A 노드 내지 D 노드에 저장된 클로토 후보 플래그의 합집합 연산(union 연산)을 수행하여, 클로토 후보 플래그를 계산할 수 있다.
- [100] 예를 들어 도 9를 참고하면 a3 생성 당시 A 노드에 저장된 프레임 정보에는 f2에 속한 루트는 {a3, d2}이고, A 노드에 추가하는 클로토 후보 플래그는 {b1, c1, d1, e1}가 된다. 이에 추가하여, c3 생성 당시 C 노드에 저장된 프레임 정보에는 f2에 속한 루트는 {a3, c3, d2}이고, 플래그 테이블의 C 노드에는 {a1, b1, c1, d1, e1}이 저장된다. 따라서, 플래그 테이블의 A 노드, C 노드 및 D 노드에 대응하는 항목에 포함된 원소들 중에서 전체 노드의 수(n)의 1/3 개 (즉, 1.67) 이상의 항목에 속하는 원소는 {a1, b1, c1, d1, e1}이 되므로, {a1, b1, c1, d1, e1}가 c3 생성 당시 C 노드에 의해 클로토 후보 플래그로 추가된다.
- [101] 이후 도 10을 참고하면, B 노드는 {a3, b2, c3, d2}들을 생성한 노드인 A 노드

내지 D 노드에 저장된 클로토 후보 플래그의 합집합인 {a1, b1, c1, d1, e1}에 대하여 클로토 후보 플래그를 추가한다. 보다 구체적으로 a3 생성 시점에서 A 노드는 a1을 제외한 모든 f1에 속한 루트들인 {b1, c1, d1, e1}를 CC로 체크할 수 있다. 그리고 c3 생성 시점에서 C 노드는 a1을 포함한 f1의 모든 루트들인 {a1, b1, c1, d1, e1}를 CC로 체크할 수 있다. 한편, b3는 A 노드의 a3와 C 노드의 c3를 참조하기 때문에 a3 생성 시점과 c3 생성 시점의 CC 목록을 전달 받을 수 있다. B 노드는 b3를 만들고 a3와 c3를 연결한 시점에 a3와 c3의 CC 목록에 대해 유니온 연산을 수행할 수 있다. 이에 따라 B 노드는 b3 생성 시점에서 f1의 모든 루트인 {a1, b1, c1, d1, e1}에 대해 CC 항목으로 체크할 수 있다. 실시 예에서 유니온 연산을 하는 이유는 b3에서는 a3와 c3에 도달 가능한 모든 이벤트 블록에 도달할 수 있기 때문이다. 즉, b3는 a3와 c3의 모든 도달 정보를 포함하여 도달할 수 있다. 이외의 실시 예에서도 이와 같은 방식으로 CC 항목에 대한 체크를 수행할 수 있다.

- [102] 또한 실시 예에서 CC 목록과 관련해서 n+1번째 프레임에 속한 루트에 대응하는 이벤트 블록에서 n번째 프레임에 속하는 루트들을 체크할 수 있다. 따라서 도 9 및 10의 실시 예에서 f1에 속한 루트들에 대해 f2에 속한 루트들의 공통도달 수를 기반으로 CC 목록이 체크될 수 있다. 또한 특정 노드에서 새로운 프레임에 대한 CC리스트는 n+1번째 프레임의 루트가 생성되는 시점에 n번째 프레임에 대한 CC 리스트를 추가하는 형식으로 수행될 수 있다. 일 예로 도 10에서 b3는 f3에 속하는 루트가 되고, b3가 생성되는 시점에 대응하여 B 노드는 f2에 대한 CC 목록을 추가할 수 있다.
- [103] 한편, B 노드는 b3가 참조하는 a3 및 c3에 관한 정보를 업데이트하기 때문에, 이벤트 블록 체인에 관한 정보에 a3 및 c3가 참조하는 이벤트 블록에 관한 정보가 저장될 수 있다.
- [104] 실시 예에서 각 노드가 이벤트 블록을 생성할 때 다른 노드와 클로토 체크 목록에 대한 정보를 교환하여 이를 기반으로 클로토 여부를 체크할 수도 있다. 또한 위에서 설명한 바와 같이 플래그 테이블 정보를 기반으로 클로토 체크 목록을 업데이트 할 수도 있다.
- [105] 도 11은 일 실시예에 따른 노드가 클로토를 결정하는 방법을 설명하기 위한 도면이다.
- [106] 도 4는 일 실시예에 따른 노드가 도 7 내지 도 10에 설명된 방법에 따라 프레임 f4에 해당하는 이벤트 블록을 생성한 경우의 이벤트 블록 체인을 나타낸다. 일 실시예에 따른 노드는 3 프레임 이전의 프레임에 속하는 루트들에 대하여 클로토 인지 여부를 결정할 수 있다. 도 4를 참고하면, 프레임 번호가 4인 f4에 속하는 루트들이 생성될 때, 3 프레임 이전인 f1에 해당하는 루트들(410)이 클로토인지 여부를 결정할 수 있다.
- [107] 도 11은 A 노드가 a5, c5, d4를 참조하여 새로운 이벤트 블록 a6을 생성한 것을 가정할 때, A 노드에 저장된 데이터(1110)를 나타낸다. A 노드에 저장된 플래그

테이블에서 A 노드에 대응하는 항목에는 프레임 f3에 해당되면서, a6로부터 도달 가능한 루트들인 {a4, b3, c4, d3, e3}가 저장될 수 있다. 이때 플러그 테이블의 A 노드에 대응하는 항목에 저장된 루트들의 수는 총 5 개로, 분산 데이터 시스템에 포함된 노드 수($n=5$)의 $2/3$ 보다 큰 값이다($5 > 3.33$). 따라서, A 노드는 a6를 새로운 루트로 결정할 수 있으며, f3 항목에는 이미 A 노드의 루트인 a4가 저장되어 있으므로, 새로운 루트 a6은 새로운 프레임 f4 항목에 저장될 수 있다.

- [108] 이때, A 노드는 클로토 체크 목록의 클로토 후보 정보에 기초하여, f4 보다 3 프레임 이전인 f1 에 속한 루트들이 클로토인지 여부를 결정할 수 있다. 도 11을 참고하면, f1 에 속한 루트들인 a1, b1, c1, d1 및 e1의 클로토 후보 플러그가 추가되어 있으므로, A 노드는 a1, b1, c1, d1 및 e1가 클로토인 것으로 결정할 수 있다. 일 실시예에 따른 데이터 분산 저장 방법은 클로토 체크 목록에 기초하여 클로토를 결정하기 때문에, 연산량이 크게 감소될 수 있다.
- [109] 도 12a 내지 도 12c는 일 실시예에 따른 노드가 최소 시간 값을 결정하는 방법을 설명하기 위한 도면이다.
- [110] 일 실시예에 따른 노드는 이벤트 블록을 클로토로 결정하고 난 후, 이벤트 블록의 합의 시간 값을 결정하는 데 기초가 되는 클로토의 최소 시간 값을 계산할 수 있다. 클로토의 최소 시간 값은, 노드가 클로토를 결정한 시점에 생성했던 이벤트 블록보다 1 프레임 뒤에서 새로운 루트를 생성할 때 계산될 수 있다. 이하에서 이벤트 블록의 타임스탬프 값은 램포트 타임스탬프 알고리즘에 따라 계산된 램포트 타임스탬프 값인 논리적 시간 값을 의미한다.
- [111] 도 12a는 최소 시간 값을 결정하는 방법을 나타내는 흐름도이다.
- [112] 도 12a를 참조하면, S1210에서, 노드는 클로토를 결정한 시점에 생성했던 이벤트 블록보다 1 프레임 뒤에서 새로 생성한 루트에서, 클로토를 결정한 시점에 생성했던 이벤트 블록과 동일한 프레임에 속한 루트들 중에서 도달 가능한 루트가 있는지 확인할 수 있다.
- [113] S1220에서, 노드는 확인된 루트들의 타임스탬프 값을 확인할 수 있다.
- [114] S1230에서, 노드는 확인된 타임스탬프 값들 중에서 가장 높은 빈도 수의 타임스탬프 값을 확인할 수 있다.
- [115] S1240에서, 노드는 가장 높은 빈도 수의 타임스탬프 값이 다수일 때, 상기 다수의 타임 스탬프 값 중 가장 낮은 값의 타임스탬프 값을 최소 시간 값으로 결정할 수 있다. 만약, 가장 높은 빈도 수의 타임스탬프 값이 하나라면, S1240는 생략할 수 있다.
- [116] S1250에서, 노드는 결정된 최소 시간 값으로 최소 시간 정보를 업데이트할 수 있다.
- [117] 도 12b는 도 11의 이벤트 블록 체인이 생성되고 나서 B 노드가 a6, b5, c6를 참조하여 새로운 이벤트 블록 b6을 생성한 것을 가정할 때, B 노드에 저장된 데이터(1250)를 나타낸다. 도 11에서 A 노드에 의해 f1 에 속한 루트 {a1, b1, c1,

$d1, e1\}$ 가 클로토로 결정되고, B 노드는 a6를 참조하므로 A 노드의 클로토 체크 목록에 기초하여 자신이 저장하고 있던 클로토 체크 목록을 업데이트할 수 있다. 이때 B 노드는 자신이 저장하고 있던 클로토 체크 목록과 다른 노드로부터 전달받은 클로토 체크 목록을 유니온 연산(합집합)하여 클로토 체크 목록을 업데이트할 수 있다. 따라서 B 노드는 $\{a1, b1, c1, d1, e1\}$ 가 클로토로 결정된 이벤트 블록임을 알게 된다. 이후, B 노드는 최소 시간 값을 계산할 수 있다.

- [118] 도 12c를 참조하면, B 노드는 최소 시간 값을 계산하기 위해, 먼저 클로토를 결정한 시점에 생성했던 이벤트 블록이 속한 f4의 다른 루트들 중에서, b6으로부터 도달 가능한 루트들의 타임스탬프 값을 확인한다. 예를 들어 도 12b를 참고하면, B 노드는 b6 생성하고 나서 f4에 속한 $\{a6, b5, c5, d4\}$ 루트들에 도달 가능하므로, B 노드는 $\{a6, b5, c5, d4\}$ 의 타임스탬프 값을 각각 확인한다. 이후 B 노드는 타임스탬프 값들 중에서 가장 빈도 수가 많은 타임스탬프 값을 확인할 수 있다. 만약, 가장 빈도 수가 많은 타임스탬프 값이 다수이면, 그 중에서 가장 작은 타임스탬프 값을 최소 시간 값으로 결정할 수 있다. 실시 예에서 $\{a6, b5, c5, d4\}$ 의 타임스탬프 값이 각각 $\{9, 10, 10, 11\}$ 일 때, 타임스탬프 값 10이 가장 높은 빈도 수의 타임스탬프 값으로 결정될 수 있다. 따라서, 도 12b와 같이 B 노드는 f1의 최소 시간 값을 10으로 결정할 수 있다.

- [119] 도 13a 내지 도 13c는 일 실시예에 따른 노드가 합의 시간 값을 결정하는 방법을 설명하기 위한 도면이다.

- [120] 도 13a는 합의 시간 값을 결정하는 방법을 나타내는 흐름도이다.

- [121] S1310에서, 노드는 최소 시간 값을 결정한 시점에 생성했던 이벤트 블록보다 1 프레임 뒤에서 새로 생성한 루트에서, 최소 시간 값을 결정한 시점에 생성했던 이벤트 블록과 동일한 프레임에 속한 루트들 중에서 도달 가능한 루트가 있는지 확인할 수 있다.

- [122] S1320에서, 노드는 확인된 루트들을 생성한 노드들이 확인된 루트들을 생성한 시점에 계산한 최소 시간 값들을 확인할 수 있다.

- [123] S1330에서, 노드는 확인된 최소 시간 값들 중에서 $2n/3$ 초과인 빈도 수를 갖는 최소 시간 값을 합의 시간 값으로 결정할 수 있다. 여기서 n은 분산 데이터 시스템에 포함된 노드의 수를 의미한다.

- [124] S1340에서, 노드는 결정된 합의 시간 값에 기초하여 메인 체인 정보를 업데이트할 수 있다.

- [125] 도 13b는 도 12a의 이벤트 블록 체인이 생성되고 나서 각 노드가 다수 이벤트 블록을 생성하고 난 후, E 노드가 c8, d5, e5를 참조하여 새로운 이벤트 블록 e6을 생성할 시점의 이벤트 블록 체인을 나타낸다. 도 13b를 참고하면, e6은 새로운 프레임인 f6에 속하는 루트로 결정될 수 있다. f5에 속한 루트 $\{a7, c8, d5, e5\}$ 를 생성할 시점의 각 노드들은 도 12a 내지 도 12c의 방법에 따라 f1의 최소 시간 값을 계산할 수 있다. 여기서, 루트 $\{a7, c8, d5, e5\}$ 를 생성할 당시 계산된 f1의 최소 시간 값이 모두 10이라 가정하면, E 노드는 e6을 생성하고 난 후, 합의 시간

값을 결정할 수 있는데, 합의 시간 값이 결정된 이벤트 블록은 아트로포스로 지칭된다.

- [126] 도 13c는 도 13b에 기초하여, 도 13b의 E 노드가 합의 시간 값을 결정하는 방법을 설명하기 위한 도면이다.
- [127] 먼저 E 노드는 f5에 속한 루트들 중에서 도달 가능한 루트들을 생성할 당시 계산된 최소 시간 값을 확인한다. 도 13c를 참고하면, E 노드는 f5에 속한 {b6, c8, d5, e5} 루트들에 도달 가능하므로, E 노드는 {b6, c8, d5, e5} 생성 당시 계산된 최소 시간 값이 각각 10 인 것을 확인한다. 이후 E 노드는 전체 최소 시간 값의 개수에 대해 $2n/3$ 초과인 빈도 수를 갖는 최소 시간 값이 있는지 확인한다. 도 13c를 참고하면 {b6, c8, d5, e5} 생성 당시 계산된 최소 시간 값이 모두 10이므로, 합의 시간 값은 10으로 결정될 수 있다. 다시 말해, f1에 속한 루트들 중에서 클로토로 결정된 {a1, b1, c1, d1, e1}들의 합의 시간 값은 10으로 결정되고, 합의 시간 값이 결정된 {a1, b1, c1, d1, e1}은 아트로포스가 된다.
- [128] 도 14a 및 도 14b는 일 실시예에 따른 노드가 최소 시간 값의 재선택을 수행하는 방법을 설명하기 위한 도면이다.
- [129] 일 실시예에 따른 노드는 클로토를 결정하고 2 프레임이 지난 상태에서 아트로포스를 결정할 수 없을 때, 그 다음 프레임에서 최소 시간 값을 재선택하여 합의 시간 값을 결정할 수 있다. 예를 들어, 도 11에서와 같이 f4에 속한 루트를 생성할 때, f1이 클로토인지 여부를 결정하였다면 f4로부터 2 프레임이 지난 f6에서 아트로포스를 결정할 수 없을 때, f6에서 최소 시간 값을 재선택하여 합의 시간 값을 결정할 수 있다.
- [130] 도 13b의 경우 f6에 속한 e6를 생성하고 나서 합의 시간 값이 결정될 수 있었으나, 도 13b와 달리 $2n/3$ 초과인 빈도 수를 갖는 최소 시간 값이 존재하지 않을 수 있다. 이렇게 되면, f6에서는 합의 시간 값을 결정할 수 없게 된다. 만약 f6에 속한 루트들을 생성하는 시점에서 합의 시간 값을 결정하지 못한다면, 노드는 f5에 속한 루트들 중에서 도달 가능한 루트들의 타임스탬프 값 중 최소값을 최소 시간 값으로 재선택할 수 있다.
- [131] 예를 들어, 도 14a를 참조하면, E노드에서 저장하고 있는 f5에 속한 루트들의 타임스탬프가 개시된다. f6에 속한 e6를 생성할 때, 합의 시간 값을 결정할 수 없을 때, E 노드는 f5에 속한 루트들 중에서 e6이 도달 가능한 루트들이 {e5, b6, c8, d5}라면, {e5, b6, c8, d5}의 타임스탬프 값 중 최소값인 9가 f1의 최소 시간 값으로 재선택된다.
- [132] 만약 최소 시간 값이 재선택되는 경우, f7에 속하는 루트가 생성될 때 f6에 속한 루트들 중에서 도달 가능한 루트들의 최소 시간 값 중 빈도 수가 $2n/3$ 을 초과하는지 확인할 수 있다. 만약, f7에서도 합의에 도달하지 않는다면, 동일한 과정을 f8에서 반복할 수 있다. 이렇게 되면, 하나의 최소 시간 값이 합의 시간 값으로 결정될 수 있다.
- [133] 도 14b를 참조하면, A노드에서 저장하고 있는 f6에 속한 루트들의

타임스탬프가 개시된다. A 노드가 f7에 속한 a12를 생성할 때, f6에 속한 루트들 중에서 a12가 도달 가능한 루트들이 {a10, b9, c11, d8}이면, A 노드는 {a10, b9, c11, d8} 생성 당시 재선택되었던 f1의 최소 시간 값을 확인할 수 있다. 이후 A 노드는 확인된 최소 시간 값들 중에서 $2n/3$ 초과와 빈도 수를 갖는 최소 시간 값을 합의 시간으로 결정하고, 합의 시간 값이 결정된 이벤트 블록을 아트로포스로 결정할 수 있다.

- [134] 도 15는 일 실시예에 따른 합의에 도달한 이벤트 블록들 간의 순서를 결정하는 방법을 설명하기 위한 도면이다.
- [135] 일 실시예에 따른 데이터 분산 저장 방법에서, 아트로포스로 결정된 이벤트 블록이 직간접적으로 참조하는 이벤트 블록들은 동일한 합의 시간 값을 갖는다. 만약 다수의 아트로포스가 하나의 이벤트 블록을 직간접적으로 참조하는 경우, 해당 이벤트 블록의 합의 시간 값은 두 아트로포스 중에서 합의 시간 값이 이른 아트로포스의 합의 시간 값으로 결정된다.
- [136] 도 15를 참조하면, 각 이벤트 블록 내 숫자는 램포트 타임스탬프 값을 의미하며, 각 폐곡선 내의 가장 오른쪽에 위치한 이벤트 블록들(1510 내지 1550)은 아트로포스로 결정된 이벤트 블록을 나타낸다. 또한, 동일한 폐곡선에 속한 이벤트 블록들은 동일한 합의 시간 값을 갖는 이벤트 블록들을 의미한다. 한편, 도 15의 제1 아트로포스(1510)와 제2 아트로포스(1520) 모두가 참조하는 이벤트 블록(1560)이 제1 아트로포스(1510)와 동일한 합의 시간 값을 가지는 것을 나타낸다. 따라서, 제1 아트로포스(1510)와 제1 아트로포스(1510)가 직간접적으로 참조하는 이벤트 블록들은 제2 아트로포스(1520) 및 제2 아트로포스(1520)가 직간접적으로 참조하는 이벤트 블록보다 이른 시간에 생성된 것으로 합의된 것임을 알 수 있다.
- [137] 만약, 동일한 합의 시간 값을 갖는 두 이벤트 블록들 간 순서를 결정해야 하는 경우, 도 5에 설명된 바와 같이 두 이벤트 블록들의 램포트 타임스탬프 값을 비교하여 이른 램포트 타임스탬프 값을 갖는 이벤트 블록이 먼저 생성된 이벤트 블록으로 결정될 수 있다. 도 15를 참고하면, 제1 아트로포스(1510)와 동일한 폐곡선 내에 위치하는 이벤트 블록(1560)은 동일한 합의 시간 값을 가지게 되지만, 이벤트 블록(1560)의 램포트 타임스탬프 값이 1이므로, 제1 아트로포스(1510)의 램포트 타임스탬프 값(3)보다 더 작은 값을 가진다. 따라서, 이벤트 블록(1560)이 제1 아트로포스(1510)보다 더 이른 시간에 생성된 이벤트 블록임을 알 수 있다.
- [138] 또한 두 이벤트 블록들의 램포트 타임스탬프 값이 동일하다면, 해시 값이 이른 이벤트 블록이 먼저 생성된 이벤트 블록으로 결정될 수 있다.
- [139] 도 16은 일 실시예에 따른 노드가 플래그 테이블을 이용하여 참조 이벤트 블록을 결정하는 방법을 설명하기 위한 도면이다.
- [140] 일 실시예에 따른 노드는 이벤트 블록을 생성할 때, 플래그 테이블을 이용하여 참조 이벤트 블록을 결정할 수 있다. 만약 참조 이벤트 블록의 수를 k라 할 때,

노드는 가장 최근에 자신이 생성한 이벤트 블록은 참조해야 하고 따라서 $k-1$ 개의 참조 이벤트 블록을 결정할 수 있다. 아래 예시들에서 k 가 3인 것을 가정하였으나, k 는 분산 데이터 시스템의 성능 및 요구조건 등에 따라 달라질 수 있다.

- [141] 도 16은 A 노드가 새로운 이벤트 블록 a_3 의 참조 이벤트 블록을 선택하기 이전에 A 노드에 저장된 플래그 테이블(1610)을 나타낸다. 여기서 A 노드에서 가장 최근에 생성된 이벤트 블록은 a_2 이고, a_2 는 D 노드 및 E 노드에서 생성한 루트를 참조하지 않기 때문에, A 노드의 플래그 테이블의 D 노드 및 E 노드에 대응하는 항목에는 기초 이벤트 블록(d_1, e_1)에 관한 정보만 저장되어 있다. 이때, A 노드는 D 노드 및 E 노드에서 생성한 루트(d_1, e_1)를 참조 이벤트 블록으로 결정함으로써 다른 노드들에 대한 정보를 빠르게 업데이트하고, 최대한 많은 노드들이 합의 알고리즘에 참여할 수 있게 할 수 있다.
- [142] 만약 참조하지 않는 노드의 수가 $k-1$ 보다 많은 경우, 노드는 참조하지 않는 노드들 중에서 임의로 참조 이벤트 블록을 $k-1$ 개 선택할 수 있다. 예를 들어, k 가 2라고 가정할 때 도 16을 참고하면 참조하지 않는 노드의 수는 2로, $k-1$ 보다 크다. 따라서, A 노드는 a_3 의 참조 이벤트 블록으로 d_1 및 e_1 중 임의로 하나를 선택할 수 있다.
- [143] 반대로 참조하지 않는 노드의 수가 $k-1$ 보다 작은 경우, 노드는 먼저 참조 이벤트 블록으로서 참조하지 않는 노드가 생성한 루트를 결정한 후, 나머지 참조 이벤트 블록은 다른 노드들이 생성한 임의의 이벤트 블록으로 결정할 수 있다. 예를 들어, k 가 4라고 가정할 때, 도 16을 참고하면 참조하지 않는 노드의 수는 2로, $k-1$ 보다 작다. 따라서, A 노드는 a_3 의 참조 이벤트 블록으로 d_1 및 e_1 를 선택하고 나서, b_1 및 c_1 중 임의로 하나를 참조 이벤트 블록으로 더 선택할 수 있다.
- [144] 본 개시의 일 실시예에 따른 노드는 다른 정보를 필요로 하지 않고, 플래그 테이블을 활용하여 참조 이벤트 블록을 결정할 수 있기 때문에, 노드의 부하를 감소시킬 수 있다. 또한, 최대한 많은 노드들이 합의 알고리즘에 참여할 수 있게 하여, 합의에 도달하는 시간을 단축시킬 수 있다.
- [145] 도 17은 일 실시예에 따른 노드가 높이 벡터 및 진입 차수 벡터를 이용하여 참조 이벤트 블록을 결정하는 방법을 설명하기 위한 도면이다.
- [146] 일 실시예에 따른 노드는 이벤트 블록을 생성할 때, 높이 벡터 및 진입 차수 벡터 값에 기초하여, 참조 이벤트 블록을 결정할 수 있다. 구체적으로, 노드는 비용함수(C_f)를 계산할 수 있으며, C_f 는 진입 차수 벡터를 높이 벡터로 나눈 값으로 정의된다($C_f=I/H$).
- [147] 도 17은 A 노드가 a_3 이벤트 블록을 생성할 때, A 노드에 저장되는 높이 벡터(1710)와 진입 차수 벡터(1720)를 나타낸다. 높이 벡터(1710)와 진입 차수 벡터(1720)를 기초하여 계산된 비용함수(1730)에 따르면, 비용함수 값이 가장 낮은 노드는 D 노드 및 E 노드이므로, A 노드는 a_3 의 참조 이벤트 블록으로 d_2 와

e1을 결정할 수 있다. 여기서 D 노드가 가장 최근에 생성한 이벤트 블록은 d2이지만, A 노드는 d2를 알 수 없으므로, 기초 이벤트 블록인 d1만 있다고 가정하여, D 노드의 높이 벡터 값을 1로 한다. 상술한 비용함수에 기초로 한 참조 이벤트 블록을 결정하는 방법은 정밀하게 참조 이벤트 블록을 결정할 수 있는 효과가 있다.

- [148] 본 개시의 합의 알고리즘에 따르면, 상술된 참조 이벤트 결정 방법 이외에도 다양한 방법을 통해 참조 이벤트 블록이 결정될 수 있으며, 참조 이벤트 결정 방법은 이에 제한되지 않는다.
- [149] 도 18a 내지 도 18d는 일 실시예에 따른 노드가 다른 노드와 데이터를 송수신하는 방법을 설명하기 위한 도면이다.
- [150] 분산 데이터 시스템에 포함된 노드들이 동일한 데이터를 가질 수 있도록 하기 위해, 일 실시예에 따른 노드는 참조 이벤트 블록을 생성한 노드로부터 수신된 데이터에 기초하여, A 노드에 저장된 데이터들을 업데이트한다.
- [151] 도 18a는 A 노드가 B 노드의 이벤트 블록을 참조하여 새로운 이벤트 블록을 생성할 때, 노드 간에 데이터를 송수신하는 절차를 나타내는 도면이다.
- [152] 먼저 S1810에서, A 노드는 B 노드에게 데이터를 요청할 수 있다. 예를 들어, A 노드는 새로운 이벤트 블록을 생성하면서 B 노드로부터 생성된 이벤트 블록을 참조하고자 할 때 B 노드에게 데이터를 요청할 수 있다. 이때, A 노드는 자신이 가진 데이터 중 적어도 일부를 데이터 요청과 함께 B 노드에 전달할 수 있다.
- [153] 이후 S1820에서 B 노드는 데이터 요청에 응답하여, A 노드에게 데이터를 전달할 수 있다. 실시 예에서 B 노드는 저장된 데이터 모두를 A 노드에 전달할 수 있다. 또 다른 실시 예에서 B 노드는 A 노드로부터 수신된 데이터와 자신의 데이터를 비교하여, 상이한 데이터만 A 노드에 전달할 수 있다. 이러한 경우, A 노드와 B 노드 간 송수신되는 데이터의 양이 감소되는 효과를 얻을 수 있다. 실시 예에서 B 노드가 A 노드에 전달하는 데이터는 도 6에서 설명한 정보 중 적어도 하나가 포함될 수 있다.
- [154] S1830에서, A 노드는 B 노드로부터 수신된 데이터에 기초하여 자신의 데이터를 업데이트할 수 있다.
- [155] 도 18b 내지 도 18c는 A 노드가 b3 및 c2를 참조하여 새로운 이벤트 블록 a4를 생성할 때 데이터 절차를 설명하기 위한 도면이다. A 노드는 도 18a에 기재된 절차를 B 노드 및 C 노드와 각각 수행한 후, B 노드 및 C 노드로부터 수신된 데이터를 이용하여 A 노드의 데이터를 업데이트할 수 있다.
- [156] 도 18b를 참고하면, A 노드가 B 노드에게 데이터 요청과 함께 a4를 생성할 당시 A 노드가 알고 있는 이벤트 블록 체인(1840)에 관한 정보 중에서, B 노드가 가장 최근에 생성한 b2와 b2가 참조하는 이벤트 블록들(1850)에 관한 정보를 B 노드에 전달할 수 있다. 여기서, B 노드가 가장 최근에 생성한 이벤트 블록은 b3이지만 A 노드는 b3의 존재를 알지 못하기 때문에, A 노드는 b3에 관한 정보를 B 노드에게 전달할 수 없다.

- [157] B 노드는 A 노드로부터 b2와 b2가 참조하는 이벤트 블록들(1850)에 관한 정보를 수신한 후, B 노드는 B 노드가 가장 최근에 생성한 이벤트 블록인 b3 및 b3가 참조하는 이벤트 블록들에 관한 정보를 A 노드에게 전달할 수 있다.
- [158] 도 18c는 B 노드가 A 노드에게 전달하는 b3 및 b3가 참조하는 이벤트 블록들(1860)을 나타낸다. A 노드는 B 로부터 데이터를 수신한 후, A 노드에 저장되어 있던 이벤트 블록 체인 정보, 프레임 정보, 플러그 정보 등을 업데이트할 수 있다.
- [159] 한편 도 18d는 C 노드가 A 노드의 데이터 요청에 응답하여 전달하는 이벤트 블록 체인(1870)을 나타낸다. A 노드는 데이터 요청과 함께, 도 18b에 기재된 이벤트 블록 체인(1850)에 관한 정보를 C 노드에게 전달한다. 이후, C 노드는 C 노드가 가장 최근에 생성한 이벤트 블록인 c2 및 c2가 참조하는 이벤트 블록들(1870)에 관한 정보를 A 노드에게 전달할 수 있다. 이때, A 노드가 C 노드로 전달한 이벤트 블록 체인(1850)과 C 노드가 A 노드에 전달한 이벤트 블록 체인(1870)은 서로 동일하므로, A 노드는 A 노드에 저장되어 있던 데이터를 업데이트하지 않을 수 있다.
- [160] 도 19a 및 도 19b는 일 실시예에 따른 노드가 일부 데이터를 가지치기(pruning)하는 방법을 설명하기 위한 도면이다.
- [161] 일 실시예에 따른 노드는 저장된 데이터 양을 줄이기 위하여, 일정 조건이 만족하는 데이터는 가지치기(pruning)를 할 수 있다. 예를 들어, 노드는 클로토로 결정된 루트에 관한 프레임 정보를 가지치기할 수 있다. 또한, 노드는 아트로포스로 결정된 루트에 관한 클로토 체크 목록 정보를 가지치기할 수 있다.
- [162] 도 19a는 도 11과 동일한 조건에서 A 노드에 저장된 데이터(1110)를 나타낸다. 다시 말해 A 노드는 플러그 테이블, 프레임 정보 및 클로토 체크 목록에 기초하여 a6를 생성하면서 a1 내지 e1이 클로토 임을 결정할 수 있다. a1 내지 e1이 클로토로 결정되면, 프레임 정보의 a1 내지 e1이 속한 f1 항목(1915)은 더 이상 필요가 없게 된다. 따라서, A 노드는 a1 내지 e1이 속한 프레임 정보의 f1 항목(1915)을 삭제할 수 있다.
- [163] 한편, 도 19b는 도 13a와 동일한 이벤트 블록 체인 상에서 E 노드에 저장된 데이터(1920)를 나타낸다. 여기서 E 노드는 이미 f1에 속한 루트뿐 아니라, f2에 속한 루트들이 클로토인 것으로 결정하였으므로, f1 및 f2에 관한 프레임 정보는 가지치기된 상태일 수 있다. 또한, E 노드는 e6를 생성하고 나서 a1 내지 e1이 아트로포스 임을 결정할 수 있다. a1 내지 e1이 아트로포스로 결정되면, a1 내지 e1에 관한 클로토 체크 목록(1925)과 최소 시간 정보(1935)는 더 이상 필요가 없게 된다. 따라서, E 노드는 a1 내지 e1이 속한 클로토 체크 목록(1925)과 최소 시간 정보(1935)를 삭제할 수 있다.
- [164] 본 개시의 합의 알고리즘에 따르면, 상술된 데이터 가지치기 방법 이외에도 다양한 방법을 통해 노드에 저장되는 데이터 양을 감소시킬 수 있으며, 데이터 가지치기 방법은 이에 제한되지 않는다.

- [165] 도 20은 일 실시예에 따른 데이터 분산 저장 방법을 설명하기 위한 흐름도이다.
- [166] 도 20을 참고하면, S2010에서 제1 노드는 제1 이벤트 블록 생성을 위한 적어도 하나의 참조 이벤트 블록을 결정할 수 있다. 만약, 제1 이벤트 블록의 참조 이벤트 블록이 2개 이상의 다른 노드에 의해 생성된 이벤트 블록들과 제1 노드가 제1 이벤트 블록 생성 이전에 생성한 이벤트 블록일 때, 제1 노드는 플래그 테이블에서 제1 노드에 대응하는 항목에 저장된 집합에 기초하여, 2개 이상의 다른 노드에 의해 생성된 이벤트 블록들을 제1 이벤트 블록의 참조 이벤트 블록으로 결정할 수 있다. 또한, 제1 노드는 높이 벡터 및 진입 차수 벡터에 기초하여 비용함수를 계산하고, 비용함수의 계산 결과에 기초하여, 2개 이상의 다른 노드에 의해 생성된 이벤트 블록들을 제1 이벤트 블록의 참조 이벤트 블록으로 결정할 수 있다. S2020에서, 제1 노드는 적어도 하나의 참조 이벤트 블록에 대응하는 시간 정보에 기초하여 제1 이벤트 블록의 시간 정보를 결정할 수 있다. 여기서, 제1 이벤트 블록의 시간 정보는 제1 이벤트 블록의 타임스탬프 값을 포함할 때 S2020은 제1 이벤트 블록의 참조 이벤트 블록의 타임스탬프 값 중에서 최대값 보다 1이 더 큰 값을 제1 이벤트 블록의 타임스탬프 값인 것으로 결정하는 단계일 수 있다. 또한, 분산 데이터 시스템에 포함된 모든 노드들에 의해 첫 번째로 생성된 이벤트 블록 각각의 타임스탬프 값은 1이고, 타임스탬프 값이 1인 이벤트 블록은 루트로 결정될 수 있다.
- [167] S2030에서, 제1 노드는 결정된 시간 정보를 기반으로 제1 이벤트 블록을 생성할 수 있다.
- [168] S2040에서, 제1 노드는 생성된 제1 이벤트 블록에 대응하는 정보를 기반으로 현재 프레임의 루트로 결정된 이벤트 블록 중 제1 이벤트 블록에서 도달 가능한 이벤트 블록의 정보를 포함하는 플래그 테이블을 업데이트할 수 있다. 또한, 플래그 테이블을 기반으로 생성된 제1 이벤트 블록에서 도달 가능한 현재 프레임의 루트로 결정된 이벤트 블록의 수가 제1 노드에서 확인한 전체 노드의 개수를 기반으로 결정된 제1 기준보다 많을 때, 제1 이벤트 블록을 새로운 프레임의 루트로 결정하고, 결정된 새로운 프레임의 루트에 기초하여, 제1 노드에 대응하는 프레임 정보를 업데이트할 수 있다. 일 실시예에서, 분산 데이터 시스템에 포함된 노드의 수가 n 일 때 제1 기준은 $2n/3$ 일 수 있다.
- [169] 또한, 프레임 정보에 포함된 루트의 제1 노드에서 확인한 전체 노드의 개수를 기반으로 결정된 제2 기준보다 많을 때, 플래그 테이블에 저장된 각 노드에 대응하는 루트의 개수가 제1 기준보다 많은 루트들의 집합을 기반으로 클로토 후보로 결정할 수 있다. 여기서 제1 노드는 3 개 프레임 이전의 루트들 중에서 클로토 후보로 결정된 이벤트 블록들을 클로토로 결정할 수 있다. 일 실시예에서, 분산 데이터 시스템에 포함된 노드의 수가 n 일 때 제2 기준은 $n/3$ 일 수 있다.
- [170] 또한, 제1 노드가 클로토를 결정한 시점에 생성했던 이벤트 블록보다 1 프레임 뒤에서 새로 생성한 루트에서 클로토를 결정한 시점에 생성했던 이벤트 블록과

동일한 프레임에 속한 루트들 중 도달 가능한 루트가 있는지 확인하고, 확인된 루트들의 타임스탬프 값을 확인할 수 있다. 이후 제1 노드는 확인된 타임스탬프 값들 중에서 가장 높은 빈도 수의 타임스탬프 값을 클로토의 최소 시간 값으로 결정하고, 결정된 최소 시간 값으로 제1 노드의 최소 시간 정보를 업데이트할 수 있다. 만약 가장 높은 빈도 수의 타임스탬프 값이 다수 개이면, 가장 높은 빈도 수의 타임스탬프 값 중에서 최소값을 최소 시간 값으로 결정할 수 있다. 이때 제1 노드는 클로토를 결정하고 난 후, 클로토가 속한 프레임에 관한 프레임 정보를 삭제할 수 있다.

- [171] 또한, 제1 노드가 최소 시간 값을 결정한 시점에 생성했던 이벤트 블록보다 1 프레임 뒤에서 새로 생성한 루트에서 최소 시간 값을 결정한 시점에 생성했던 이벤트 블록과 동일한 프레임에 속한 루트들 중 도달 가능한 루트가 있는지 확인할 수 있다. 그리고 나서 확인된 루트들을 생성한 노드들이 확인된 루트들을 생성한 시점에 계산한 최소 시간 값을 확인할 수 있다. 이후 확인된 최소 시간 값 중에서 제1 노드에서 확인한 노드의 개수를 기반으로 결정된 제3 기준을 초과하는 빈도 수를 갖는 최소 시간 값이 합의 시간 값으로 결정될 수 있고, 결정된 합의 시간 값에 기초하여 제1 노드는 제1 노드의 메인 체인 정보를 업데이트할 수 있다. 일 실시예에서, 분산 데이터 시스템에 포함된 노드의 수가 n 일 때, 제1 노드에서 확인한 노드의 개수를 기반으로 결정된 제3 기준은 $2n/3$ 일 수 있다.
- [172] 한편, 제1 노드는 클로토를 결정하고 2 프레임이 지난 후에도 합의 시간 값을 결정할 수 없을 때, 최소 시간 값을 재선택하여 합의 시간 값을 결정할 수 있다.
- [173] 또한, 제1 노드는 두 이벤트 블록의 합의 시간 값을 비교하여 더 이른 합의 시간 값을 갖는 이벤트 블록이 더 이른 시점에 생성된 이벤트 블록인 것으로 판단할 수 있다. 만약 두 이벤트 블록의 합의 시간 값이 동일한 경우, 두 이벤트 블록의 타임스탬프 값을 비교하여, 더 이른 타임스탬프 값을 갖는 이벤트 블록이 더 이른 시점에 생성된 이벤트 블록인 것으로 판단할 수 있다. 그러나, 두 이벤트 블록의 합의 시간 값과 타임스탬프 값이 동일한 경우, 두 이벤트 블록의 물리적 시간 값을 비교하여, 더 이른 물리적 시간 값을 갖는 이벤트 블록이 더 이른 시점에 생성된 이벤트 블록인 것으로 판단할 수 있다.
- [174] 또한, 제1 노드는 합의 시간 값을 결정하고 난 후, 합의 시간 값이 결정된 이벤트 블록이 포함된 프레임에 관한 클로토 체크 목록 및 최소 시간 정보를 삭제할 수 있다.
- [175] 한편, 제1 이벤트 블록은 제1 노드의 서명 정보, 결정된 시간 정보, 제1 노드가 생성된 물리적인 시간 값, 거래 정보 및 제1 노드의 참조 이벤트 블록의 해시 값을 포함할 수 있다.
- [176] 도 21은 일 실시예에 따른 분산 데이터 시스템에 포함된 노드의 구조를 설명하기 위한 블록도이다.
- [177] 일 실시예에 따른 분산 데이터 시스템은 2개 이상의 노드(2100)를 포함하며,

- 노드(2100)는 통신부(2110), 메모리(2120) 및 프로세서(2130)를 포함할 수 있다.
- [178] 통신부(2110)는 다른 노드와 데이터를 송수신할 수 있고, 메모리(2120)는 프로세서(2130)에 의해 처리되는 인스트럭션 및 데이터들을 저장할 수 있다.
- [179] 프로세서(2130)는 통신부(2110)에 의해 수신된 다른 노드에 관한 정보에 기초하여, 제1 이벤트 블록 생성을 위한 적어도 하나의 참조 이벤트 블록을 결정할 수 있다. 또한, 적어도 하나의 참조 이벤트 블록에 대응하는 시간 정보에 기초하여 제1 이벤트 블록의 시간 정보를 결정할 수 있다. 그리고 나서, 결정된 시간 정보를 기반으로 제1 이벤트 블록을 생성하고, 생성된 제1 이벤트 블록에 대응하는 정보를 기반으로 현재 프레임의 루트로 결정된 이벤트 블록 중 제1 이벤트 블록에서 도달 가능한 이벤트 블록의 정보를 포함하는 플래그 테이블을 업데이트할 수 있다.
- [180] 실시 예 전반에서 각 노드는 필요한 연산을 수행하기 이전에 분산 데이터 시스템에 포함된 노드의 수 n 을 확인하는 동작을 추가로 수행할 수 있다. 상기 n 은 고정된 값일 수 있으나, 시스템 설계에 따라 추가 또는 감소할 수 있는 바, 각 노드는 n 을 확인하고, 이후 확인된 n 값을 기반으로 관련된 연산을 수행할 수 있다. 일 실시 예에서 n 은 참조되는 이벤트 블록에 대응하는 노드로부터 수신하는 정보에 포함될 수 있다.
- [181] 본 실시예들은 컴퓨터에 의해 실행되는 프로그램 모듈과 같은 컴퓨터에 의해 실행 가능한 명령어를 포함하는 기록 매체의 형태로도 구현될 수 있다. 컴퓨터 판독 가능 매체는 컴퓨터에 의해 액세스될 수 있는 임의의 가용 매체일 수 있고, 휘발성 및 비휘발성 매체, 분리형 및 비분리형 매체를 모두 포함한다. 또한, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 모두 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함한다. 통신 매체는 전형적으로 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈, 또는 기타 전송 메커니즘을 포함하며, 임의의 정보 전달 매체를 포함한다.
- [182] 본 실시예의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 포함되는 것으로 해석되어야 한다.

청구범위

- [청구항 1] 제1 노드에서 데이터 분산 저장 방법에 있어서,
제1 이벤트 블록 생성을 위한 적어도 하나의 참조 이벤트 블록을 결정하는 단계;
상기 적어도 하나의 참조 이벤트 블록에 대응하는 시간 정보에 기초하여 상기 제1 이벤트 블록의 시간 정보를 결정하는 단계;
상기 결정된 시간 정보를 기반으로 상기 제1 이벤트 블록을 생성하는 단계; 및
상기 생성된 제1 이벤트 블록에 대응하는 정보를 기반으로 현재 프레임의 루트로 결정된 이벤트 블록 중 상기 제1 이벤트 블록에서 도달 가능한 이벤트 블록의 정보를 포함하는 플래그 테이블을 업데이트하는 단계를 포함하는 데이터 분산 저장 방법.
- [청구항 2] 제1항에 있어서,
상기 플래그 테이블을 기반으로 상기 생성된 제1 이벤트 블록에서 도달 가능한 현재 프레임의 루트로 결정된 이벤트 블록의 수가 상기 제1 노드에서 확인한 전체 노드의 개수를 기반으로 결정된 제1 기준보다 많을 때, 상기 제1 이벤트 블록을 새로운 프레임의 루트로 결정하는 단계; 및
상기 결정된 새로운 프레임의 루트에 기초하여, 상기 제1 노드에 대응하는 프레임 정보를 업데이트 하는 단계를 더 포함하는 데이터 분산 저장 방법.
- [청구항 3] 제2항에 있어서,
상기 프레임 정보에 포함된 루트의 수가 상기 제1 노드에서 확인한 전체 노드의 개수를 기반으로 결정된 제2 기준보다 많을 때, 상기 플래그 테이블에 저장된 각 노드에 대응하는 루트의 개수가 상기 제1 기준보다 많은 루트들의 집합을 기반으로 클로토 후보로 결정하는 단계; 및
현재 프레임에서 3 개 프레임 이전의 루트들 중에서 상기 클로토 후보로 결정된 이벤트 블록들을 클로토로 결정하는 단계를 더 포함하는, 데이터 분산 저장 방법.
- [청구항 4] 제3항에 있어서,
상기 클로토를 결정한 시점에 생성했던 이벤트 블록보다 1 프레임 뒤에서 새로 생성한 루트에서 상기 클로토를 결정한 시점에 생성했던 이벤트 블록과 동일한 프레임에 속한 루트들 중 도달 가능한 루트가 있는지 확인하는 단계;
상기 확인된 루트들의 타임스탬프 값을 확인하는 단계;
상기 확인된 타임스탬프 값들 중에서 가장 높은 빈도 수의 타임스탬프 값을 상기 클로토의 최소 시간 값으로 결정하는 단계; 및
상기 결정된 최소 시간 값으로 상기 제1 노드의 해당 프레임에 대한 최소

시간 정보를 업데이트하는 단계를 더 포함하고,
 상기 가장 높은 빈도 수의 타임스탬프 값이 다수 개이면, 상기 가장 높은 빈도 수의 타임스탬프 값 중에서 최소값을 최소 시간 값으로 결정하는 것인, 데이터 분산 저장 방법.

[청구항 5] 제3항에 있어서,
 상기 제1 노드는 상기 클로토 결정에 대응하여 상기 클로토가 속한 프레임에 관한 프레임 정보를 삭제하는 단계를 더 포함하는, 데이터 분산 저장 방법.

[청구항 6] 제4항에 있어서,
 상기 최소 시간 값을 결정한 시점에 생성했던 이벤트 블록보다 1 프레임 뒤에서 새로 생성한 루트에서 상기 최소 시간 값을 결정한 시점에 생성했던 이벤트 블록과 동일한 프레임에 속한 루트들 중 도달 가능한 루트가 있는지 확인하는 단계;
 상기 확인된 루트들을 생성한 노드들이, 상기 확인된 루트들을 생성한 시점에 계산한 상기 최소 시간 값을 확인하는 단계;
 상기 확인된 최소 시간 값 중에서 상기 제1 노드에서 확인한 노드의 개수를 기반으로 결정된 제3 기준을 초과하는 빈도 수를 갖는 최소 시간 값을 합의 시간 값으로 결정하는 단계; 및
 상기 결정된 합의 시간 값에 기초하여 상기 제1 노드의 메인 체인 정보를 업데이트하는 단계를 더 포함하는, 데이터 분산 저장 방법.

[청구항 7] 제6항에 있어서,
 상기 클로토를 결정하고 2 프레임이 지난 후에도 합의 시간 값을 결정할 수 없을 때, 최소 시간 값을 재선택하여 합의 시간 값을 결정하는 것인, 데이터 분산 저장 방법.

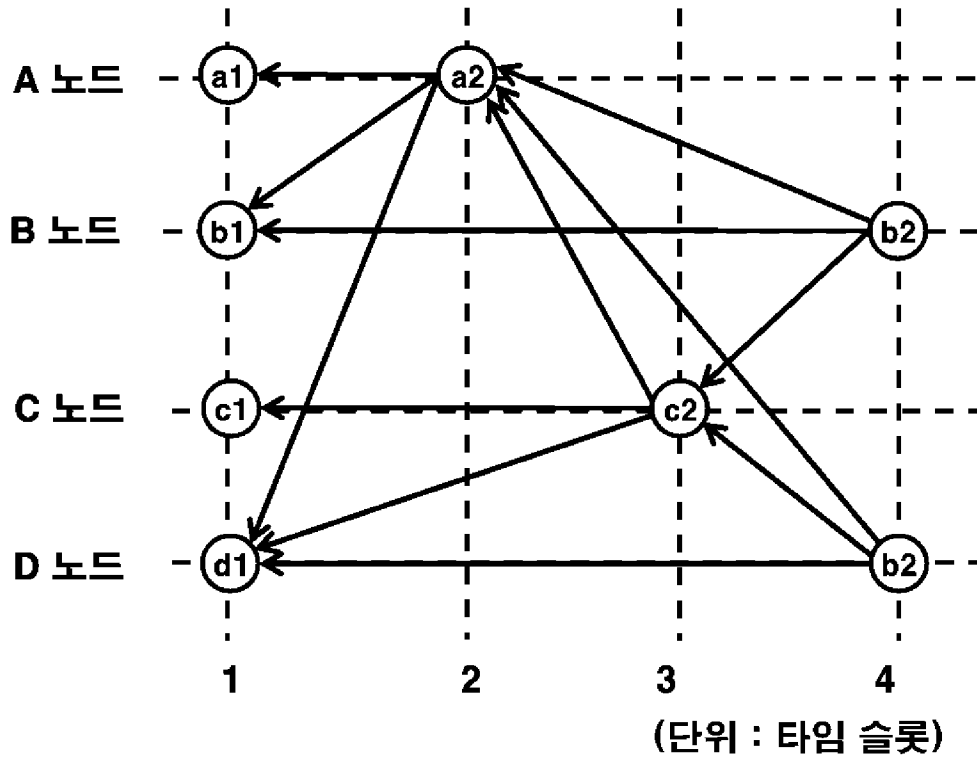
[청구항 8] 제6항에 있어서,
 두 이벤트 블록의 합의 시간 값을 비교하여 더 이른 합의 시간 값을 갖는 이벤트 블록이 더 이른 시점에 생성된 이벤트 블록인 것으로 판단하는 단계;
 상기 두 이벤트 블록의 합의 시간 값이 동일한 경우, 상기 두 이벤트 블록의 타임스탬프 값을 비교하여, 더 이른 타임스탬프 값을 갖는 이벤트 블록이 더 이른 시점에 생성된 이벤트 블록인 것으로 판단하는 단계; 및
 상기 두 이벤트 블록의 합의 시간 값과 타임스탬프 값이 동일한 경우, 상기 두 이벤트 블록의 물리적 시간 값을 비교하여, 더 이른 물리적 시간 값을 갖는 이벤트 블록이 더 이른 시점에 생성된 이벤트 블록인 것으로 판단하는 단계를 더 포함하는, 데이터 분산 저장 방법.

[청구항 9] 제6항에 있어서,
 상기 제1 노드는 상기 합의 시간 값을 결정하고 난 후, 상기 합의 시간 값이 결정된 이벤트 블록이 포함된 프레임에 관한 클로토 체크 목록 및

- 최소 시간 정보를 삭제하는 단계를 더 포함하는, 데이터 분산 저장 방법.
- [청구항 10] 제1항에 있어서,
 상기 제1 이벤트 블록은 제1 노드에 의해 생성된 것이고,
 상기 제1 이벤트 블록의 참조 이벤트 블록은 2개 이상의 다른 노드에 의해 생성된 이벤트 블록들과 상기 제1 노드가 상기 제1 이벤트 블록 생성 이전에 생성한 이벤트 블록이 되고,
 상기 제1 이벤트 블록의 참조 이벤트 블록을 결정하는 단계는,
 상기 플래그 테이블에서 상기 제1 노드에 대응하는 항목에 저장된 집합에 기초하여, 상기 2개 이상의 다른 노드에 의해 생성된 이벤트 블록들을 상기 제1 이벤트 블록의 참조 이벤트 블록으로 결정하는 단계를 포함하는, 데이터 분산 저장 방법.
- [청구항 11] 제1항에 있어서,
 상기 제1 이벤트 블록은 제1 노드에 의해 생성된 것이고,
 상기 제1 이벤트 블록의 참조 이벤트 블록은 2개 이상의 다른 노드에 의해 생성된 이벤트 블록들과 상기 제1 노드가 상기 제1 이벤트 블록의 생성 이전에 생성한 이벤트 블록이 되고,
 상기 제1 이벤트 블록의 참조 이벤트 블록을 결정하는 단계는,
 높이 벡터 및 진입 차수 벡터에 기초하여 비용함수를 계산하는 단계; 및
 상기 비용함수의 계산 결과에 기초하여, 상기 2개 이상의 다른 노드에 의해 생성된 이벤트 블록들을 상기 제1 이벤트 블록의 참조 이벤트 블록으로 결정하는 단계를 포함하는, 데이터 분산 저장 방법.
- [청구항 12] 제1항에 있어서,
 상기 제1 이벤트 블록의 시간 정보는 상기 제1 이벤트 블록의 타임스탬프 값을 포함하고,
 상기 제1 이벤트 블록의 시간 정보를 결정하는 단계는,
 상기 제1 이벤트 블록의 참조 이벤트 블록의 타임스탬프 값 중에서 최대값 보다 1이 더 큰 값을 상기 제1 이벤트 블록의 타임스탬프 값인 것으로 결정하는 단계인, 데이터 분산 저장 방법.
- [청구항 13] 제1항에 있어서,
 분산 데이터 시스템에 포함된 모든 노드들에 의해 첫 번째로 생성된 이벤트 블록 각각의 타임스탬프 값은 1이고, 타임스탬프 값이 1인 이벤트 블록은 루트로 결정되는 것인, 데이터 분산 저장 방법.
- [청구항 14] 제1항에 있어서,
 상기 제1 이벤트 블록은 상기 제1 노드의 서명 정보, 상기 결정된 시간 정보, 상기 제1 노드가 생성된 물리적인 시간 값, 거래 정보 및 상기 제1 노드의 참조 이벤트 블록의 해시 값을 포함하는 것인, 데이터 분산 저장 방법.
- [청구항 15] 2 개 이상의 노드가 포함된 분산 데이터 시스템에 있어서, 상기 노드는

다른 노드와 데이터를 송수신하는 통신부;
인struk션을 저장하는 메모리; 및
상기 인struk션을 처리하는 프로세서를 포함하고,
상기 프로세서는 제1항에 따른 방법의 단계들을 수행하는 것인, 분산
데이터 시스템.

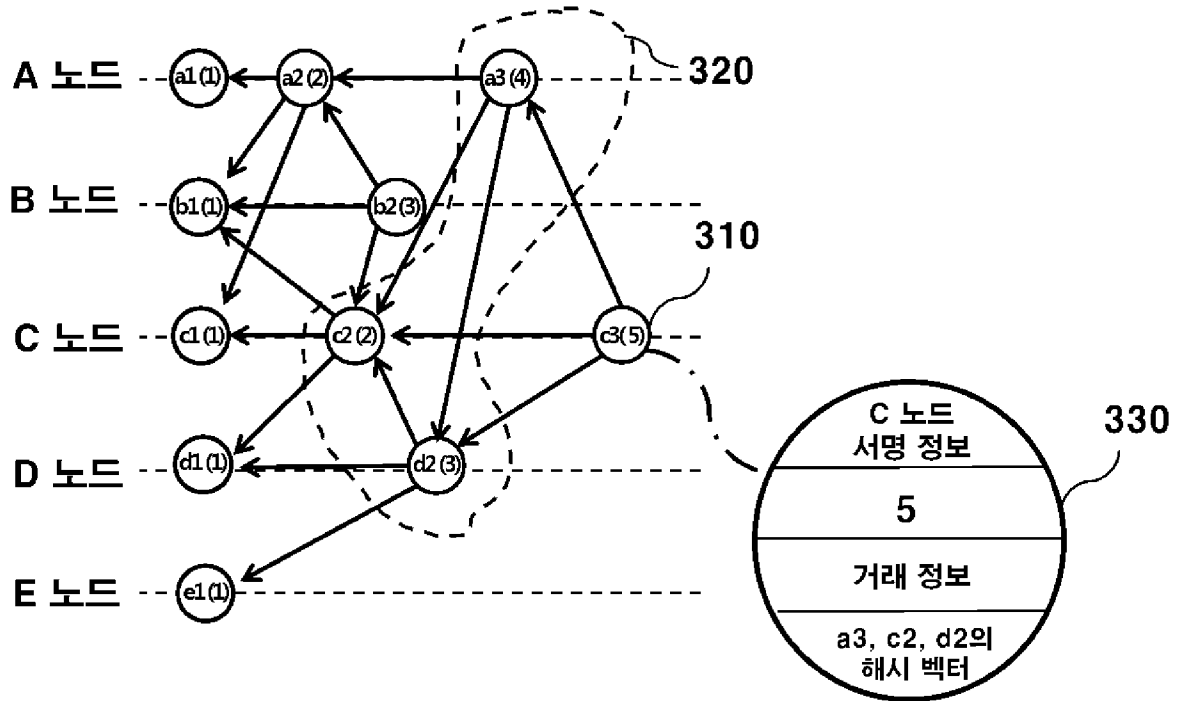
[도1]



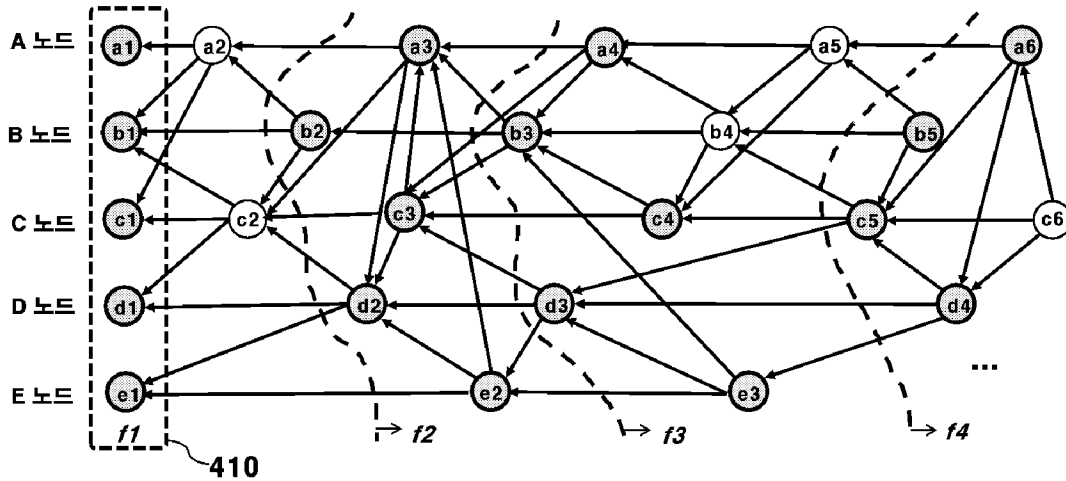
[도2]



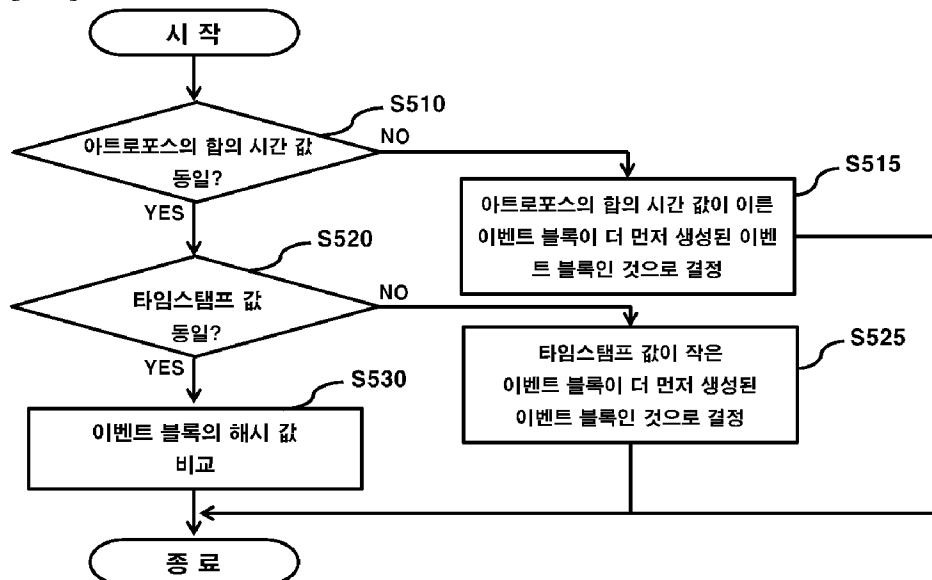
[도3]



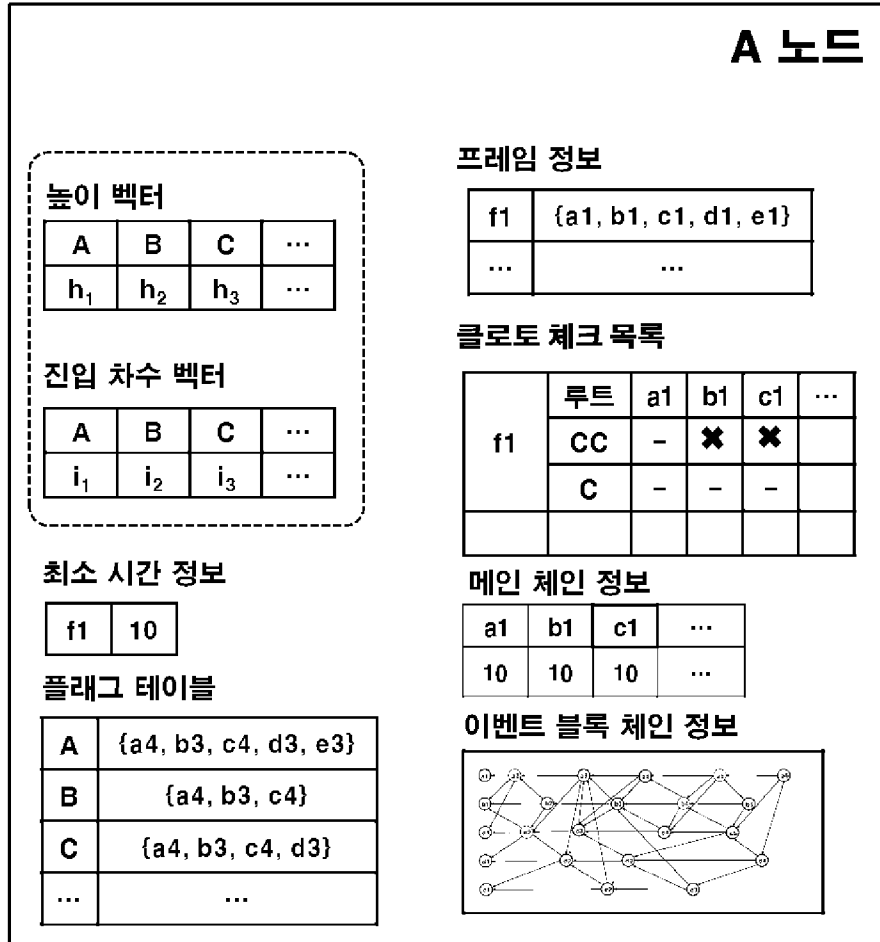
[도4]



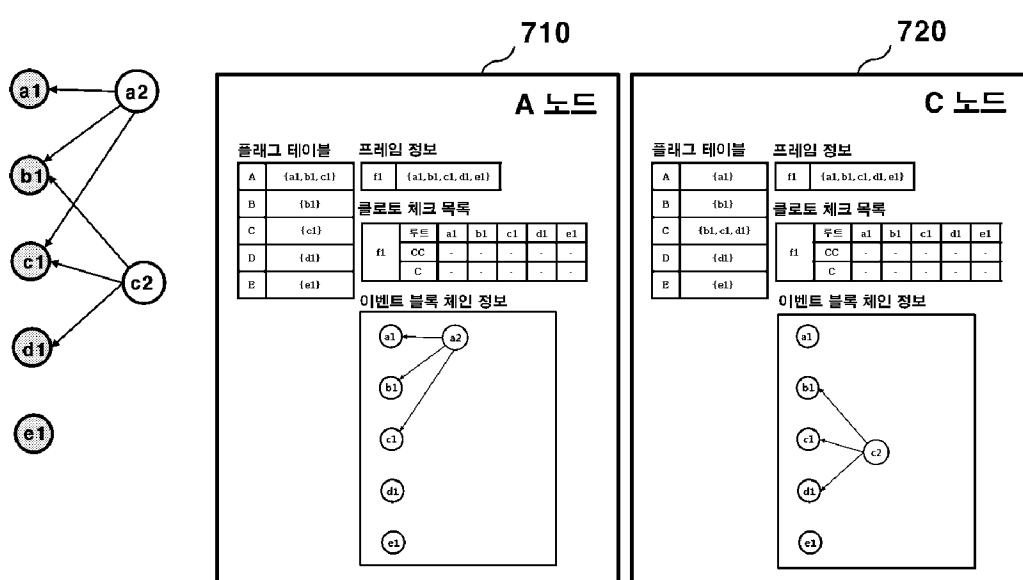
[도5]



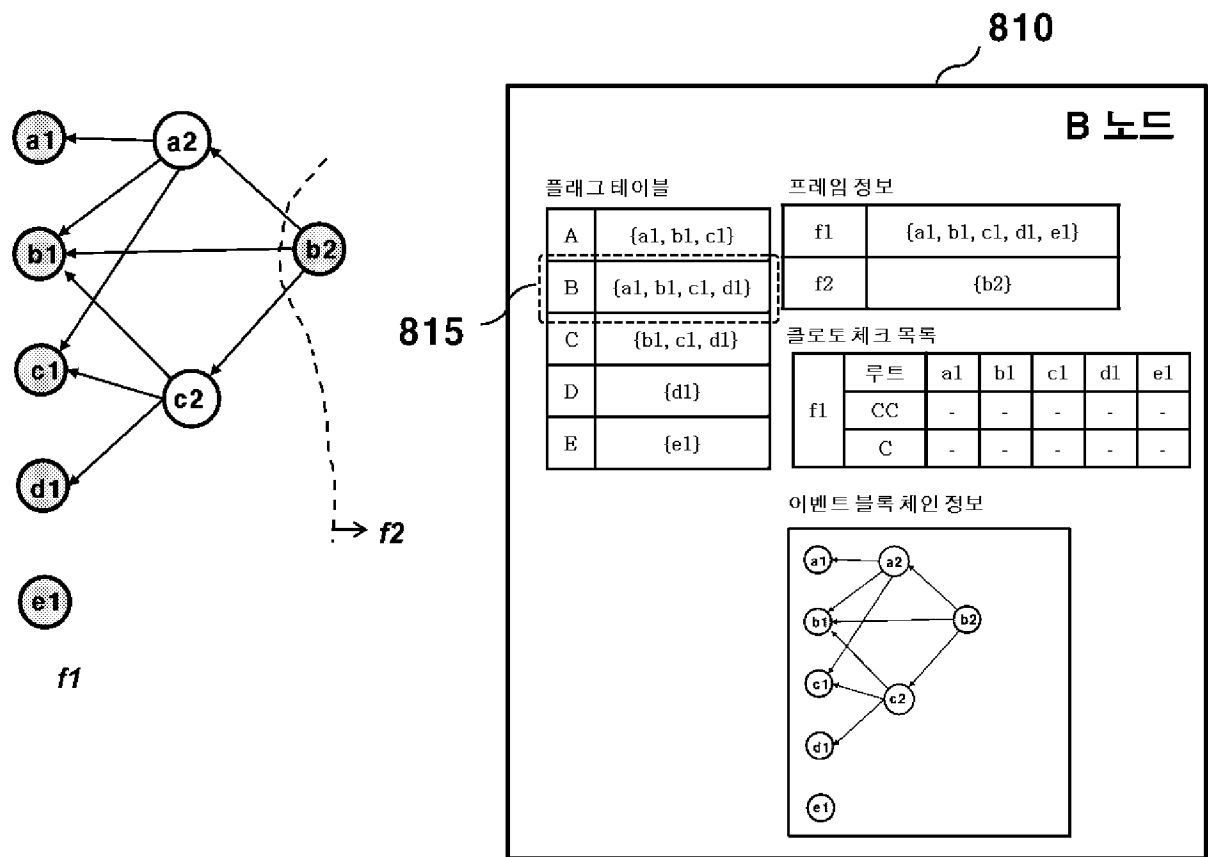
[도6]



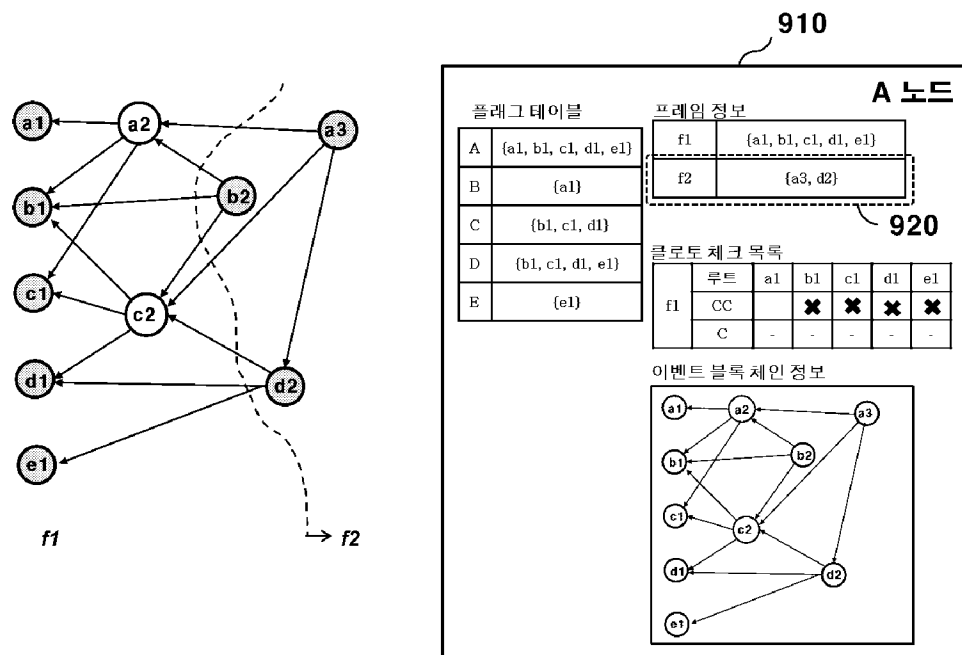
[도7]



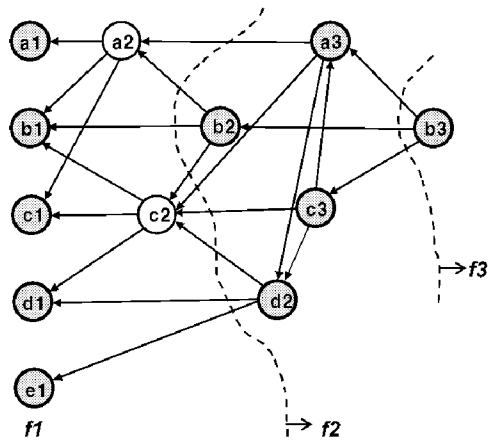
[도8]



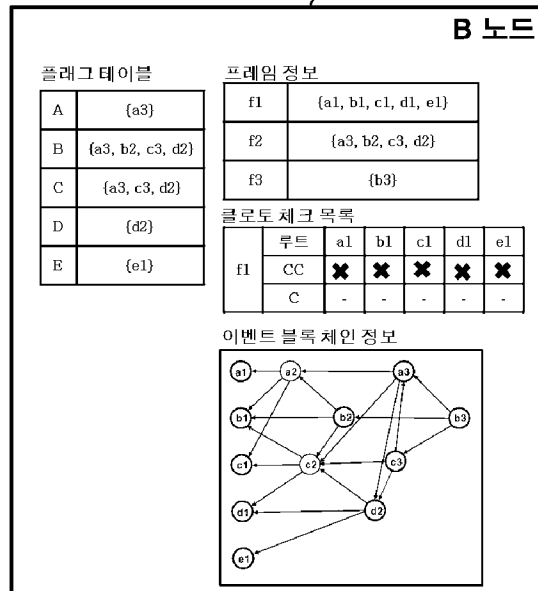
[도9]



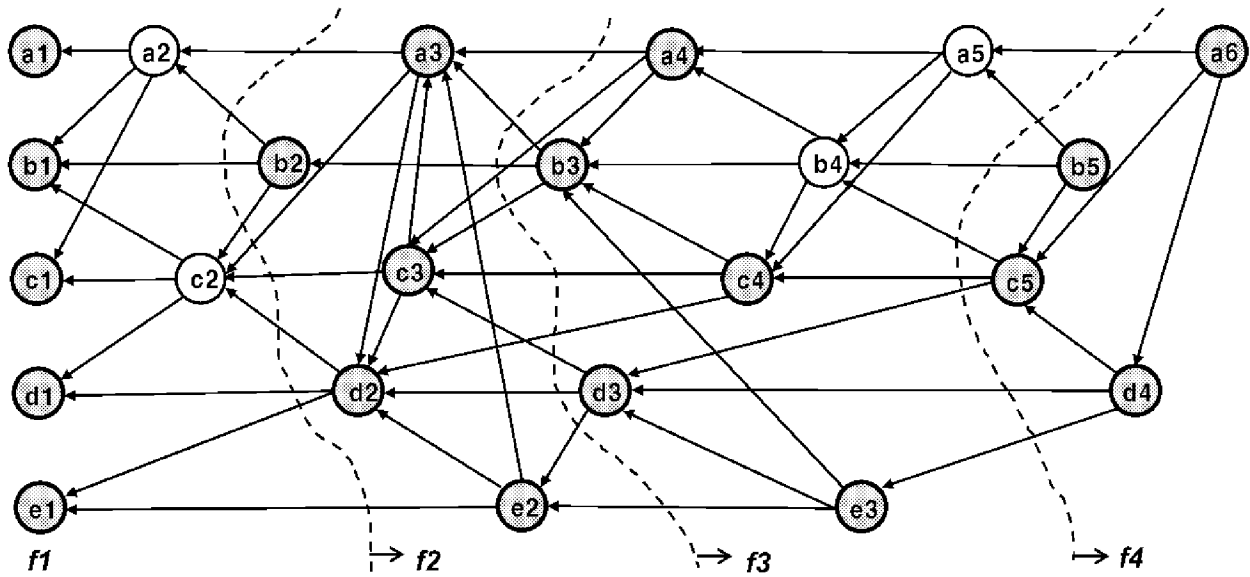
[도10]



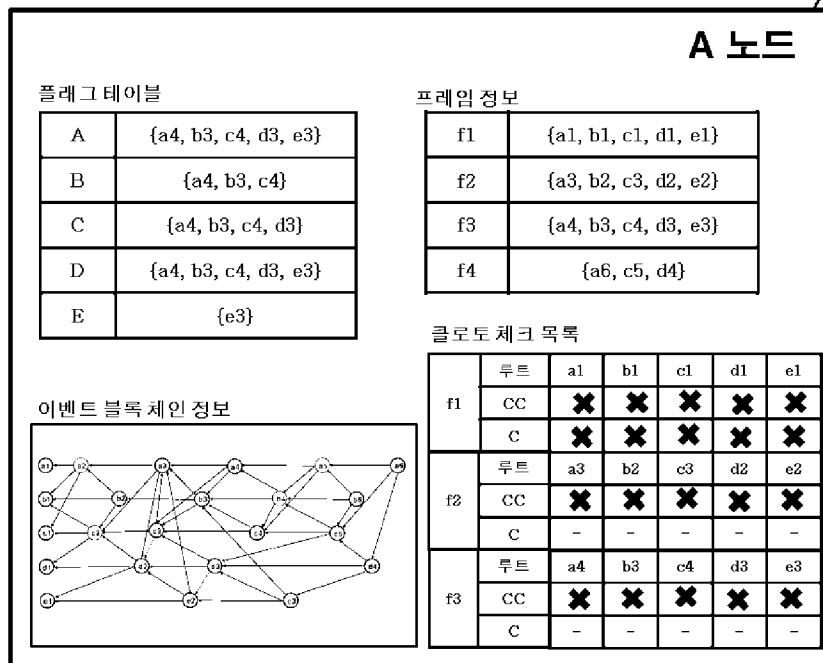
1010



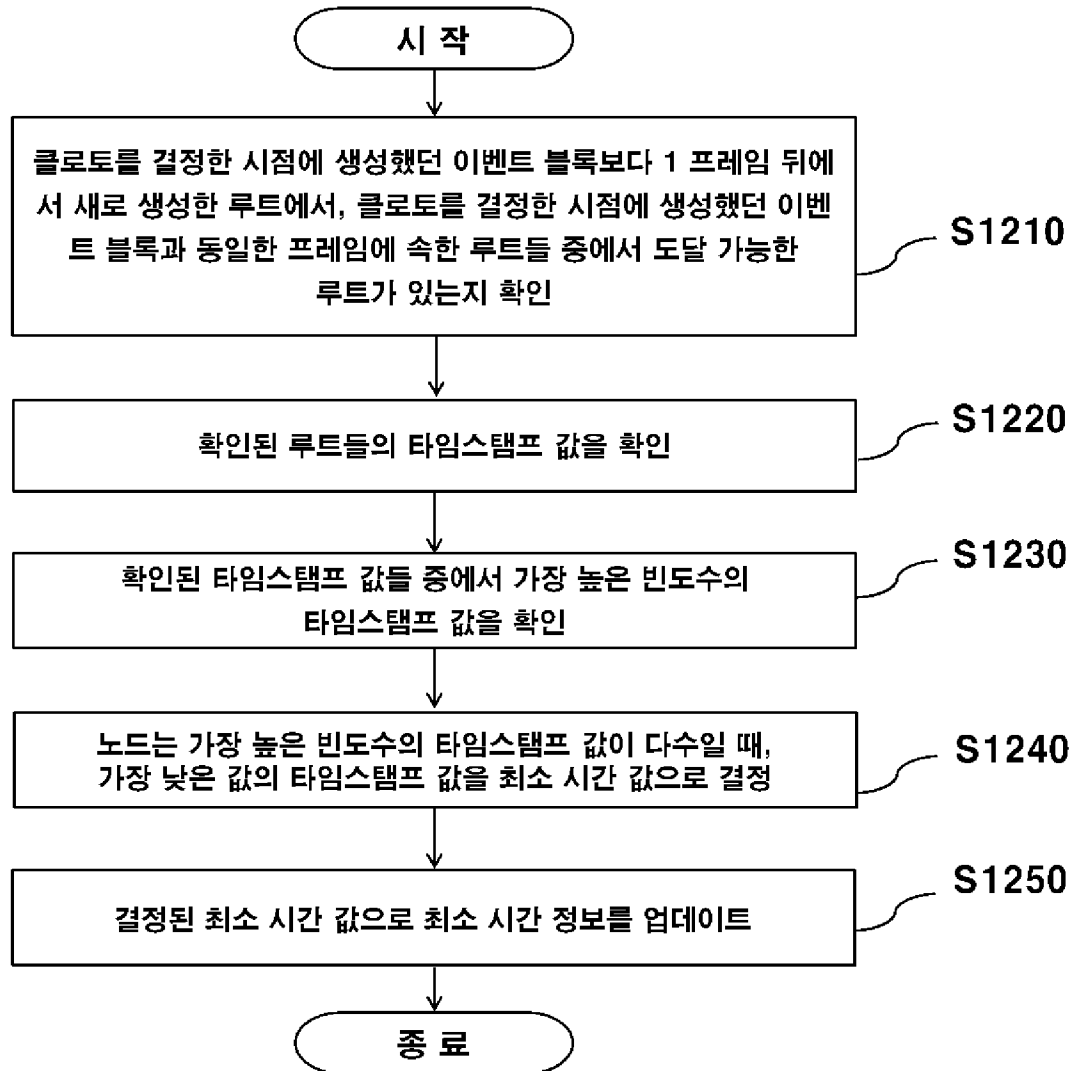
[도11]



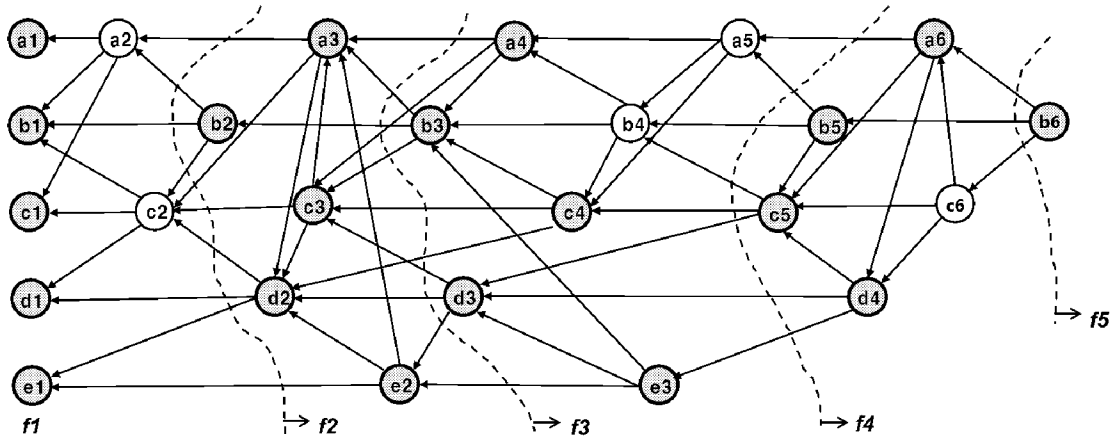
1110



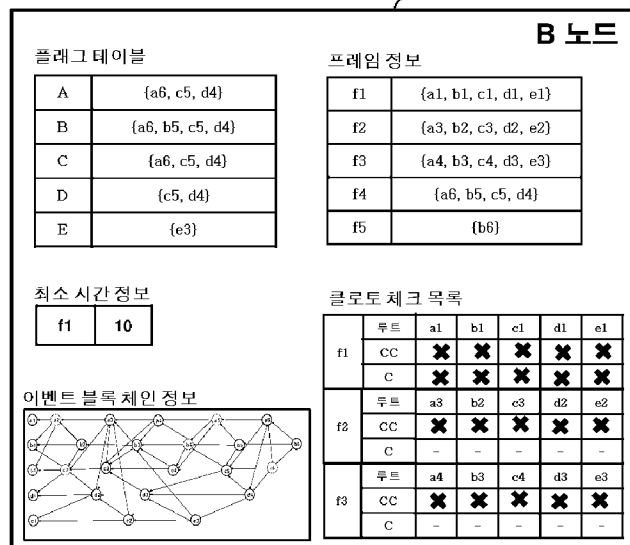
[도12a]



[도12b]



1250



[도12c]

f4에 속한 루트들 중에서 b6이 도달 가능한 루트

B	a6	b5	c5	d4
----------	----	----	----	----



----- 각 루트의 논리적 타임스탬프 값 확인

f4에 속한 루트들의 타임스탬프

9	10	10	11
---	----	----	----

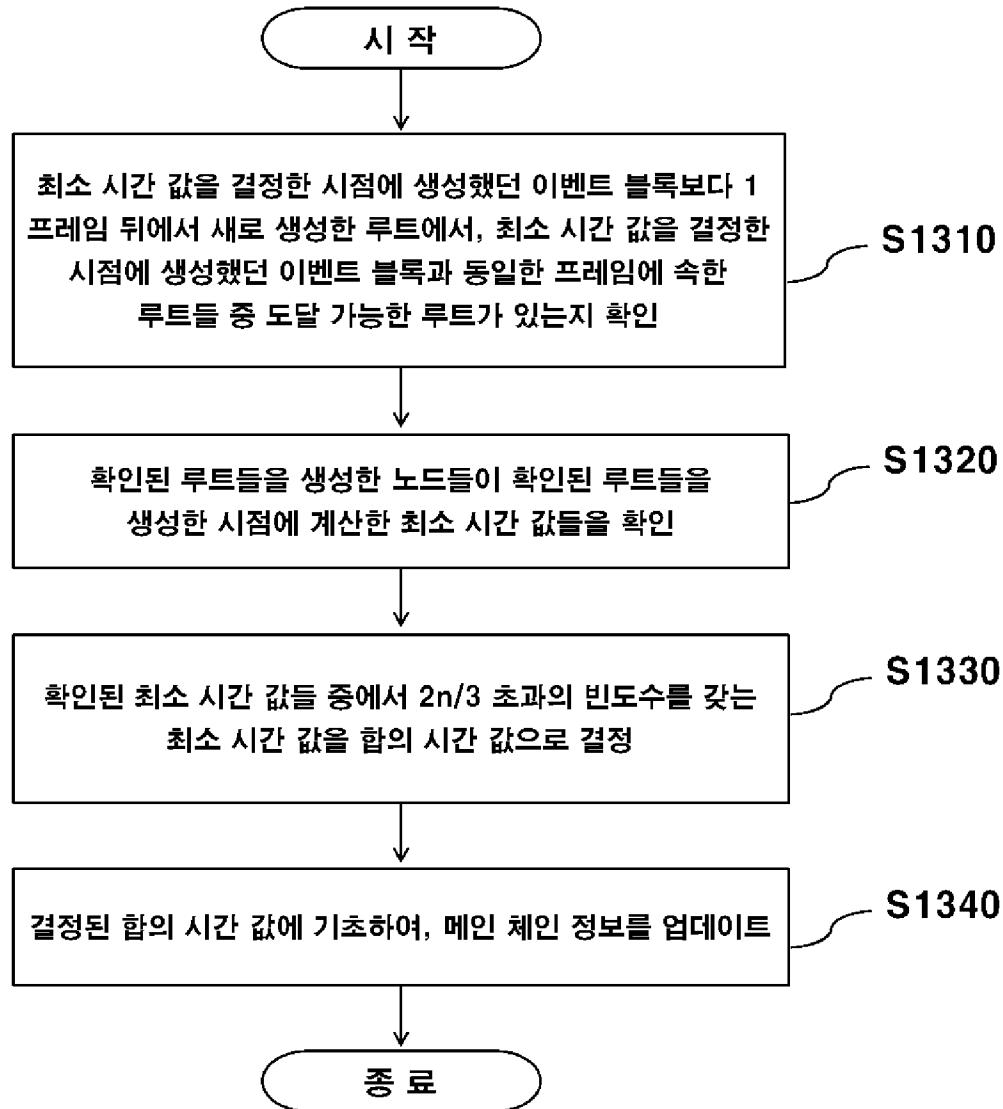
----- 가장 높은 빈도수의
타임스탬프 값 확인

10	10
----	----

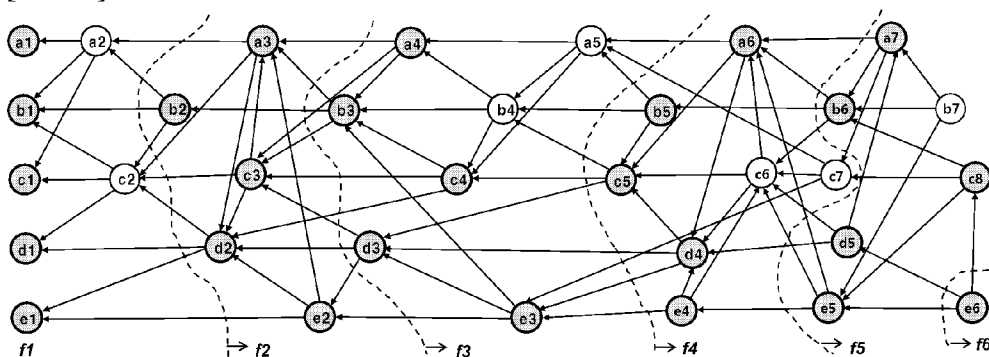
----- 가장 낮은 값의 타임스탬프 값을
최소 시간 값으로 결정

10

[도13a]



[도13b]



[도13c]

f5에 속한 루트들 중에서 e60이 도달 가능한 루트

E	e5	b6	c8	d5
---	----	----	----	----



f5에 속한 루트 생성 당시 계산된 f1의 최소 시간 값

10	10	10	10
----	----	----	----

----- 2n/3 초과인 빈도수를
갖는 최소 시간 값 확인

10	10
----	----



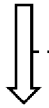
----- 합의 시간 저장

10

[도14a]

f5에 속한 루트들 중에서 e60이 도달 가능한
한 루트의 타임스탬프 값 확인

E	b6	c8	d5	e5
	11	9	15	10

----- 최소값을
최소 시간 값으로 재선택

9

[도14b]

f6에 속한 루트들 중에서 a120이 도달 가능한 루트

A	a10	b9	c11	d8
---	-----	----	-----	----



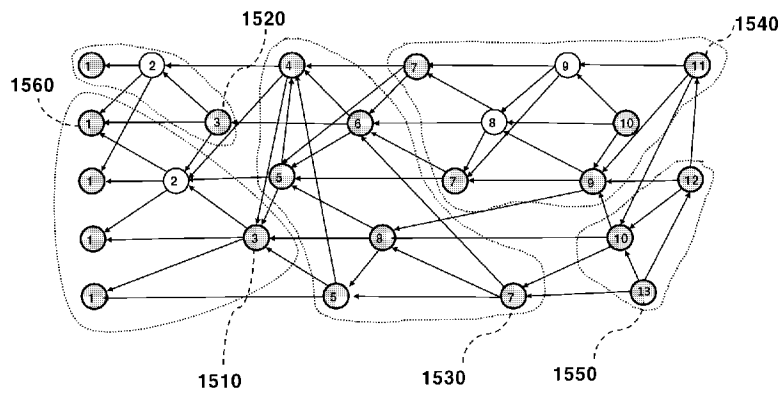
f5에 속한 루트 생성 당시 재선택된 f1의 최소 시간 값

8	8	8	4
---	---	---	---

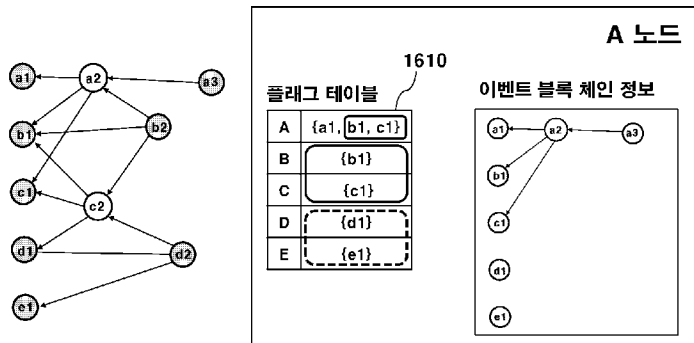
----- 2n/3 초과인 빈도수를
갖는 최소 시간 값을
합의 시간으로 결정

8

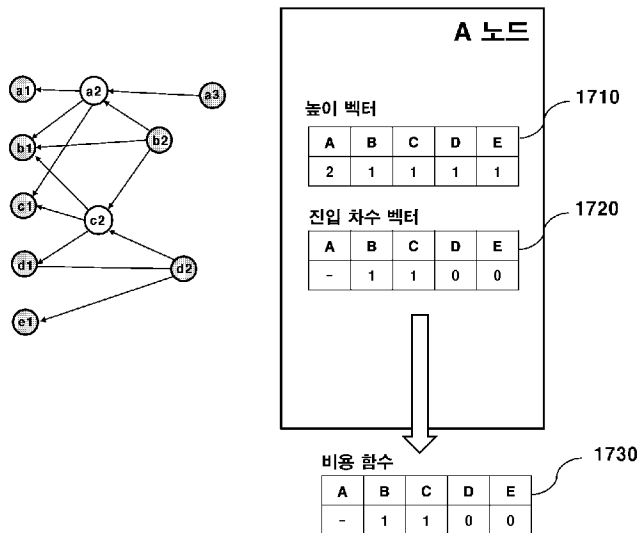
[도15]



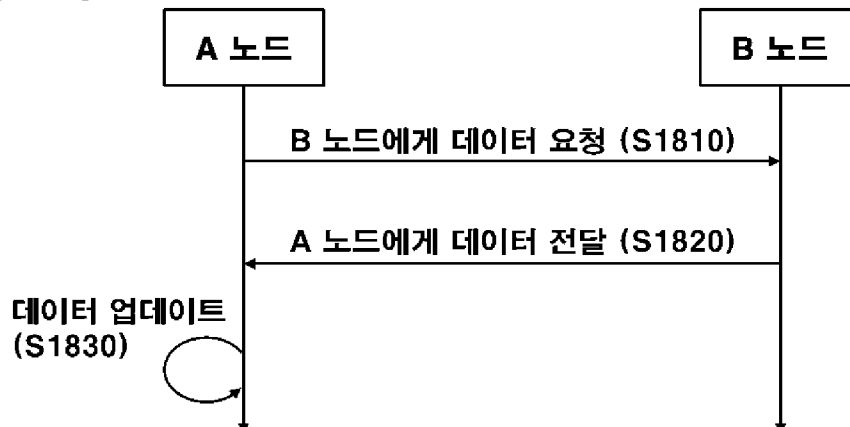
[도16]



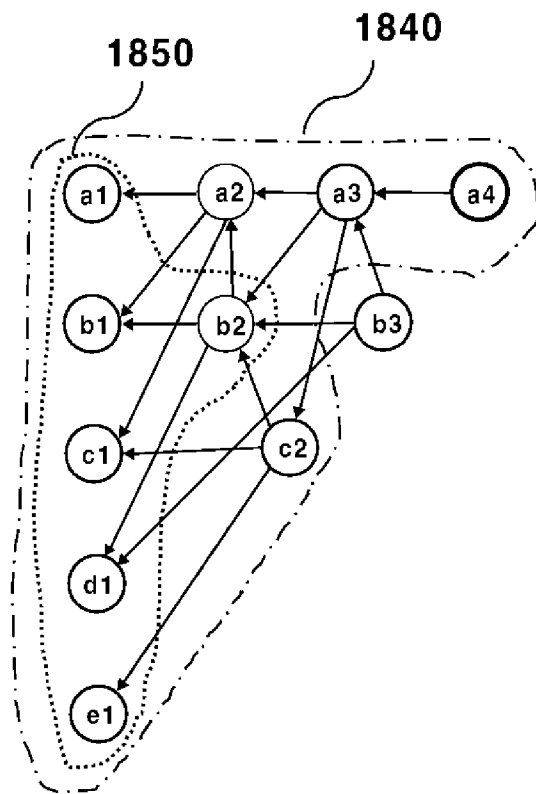
[도17]



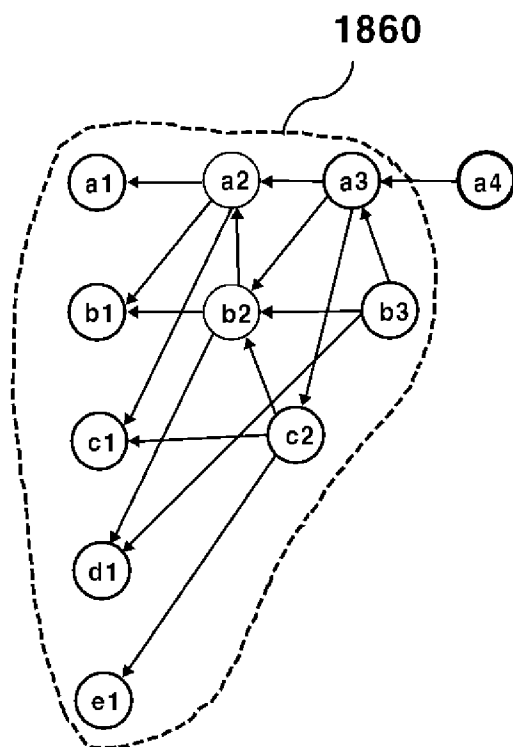
[도18a]



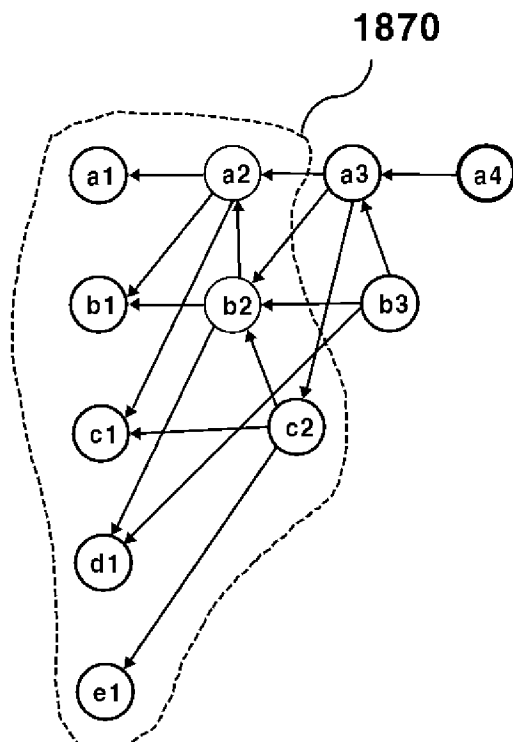
[도 18b]



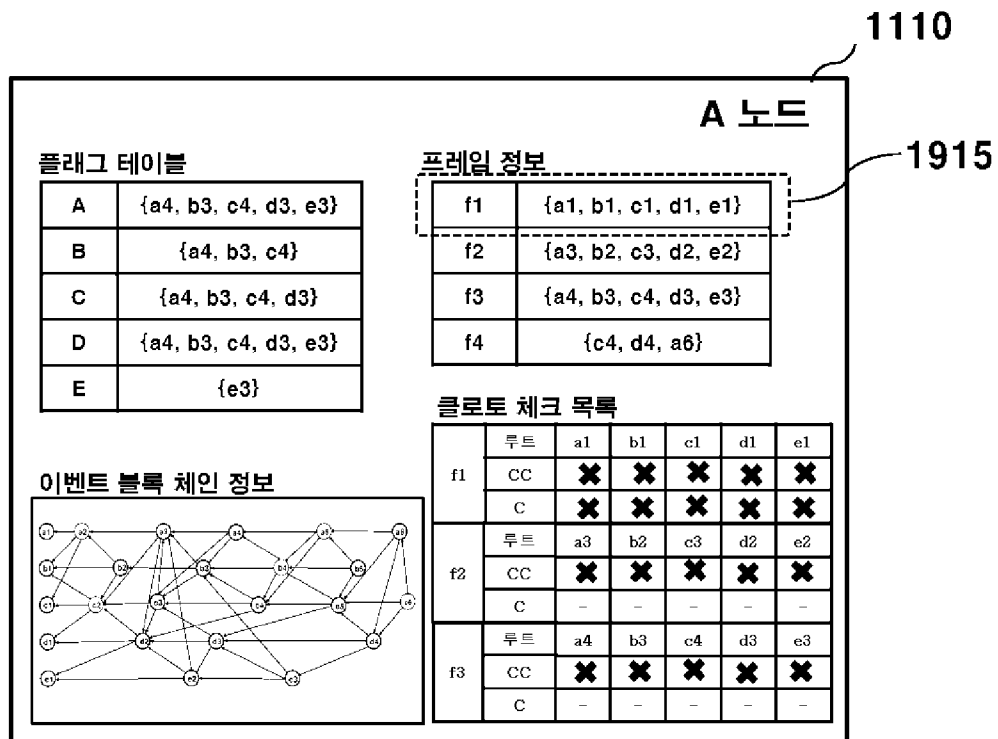
[도 18c]



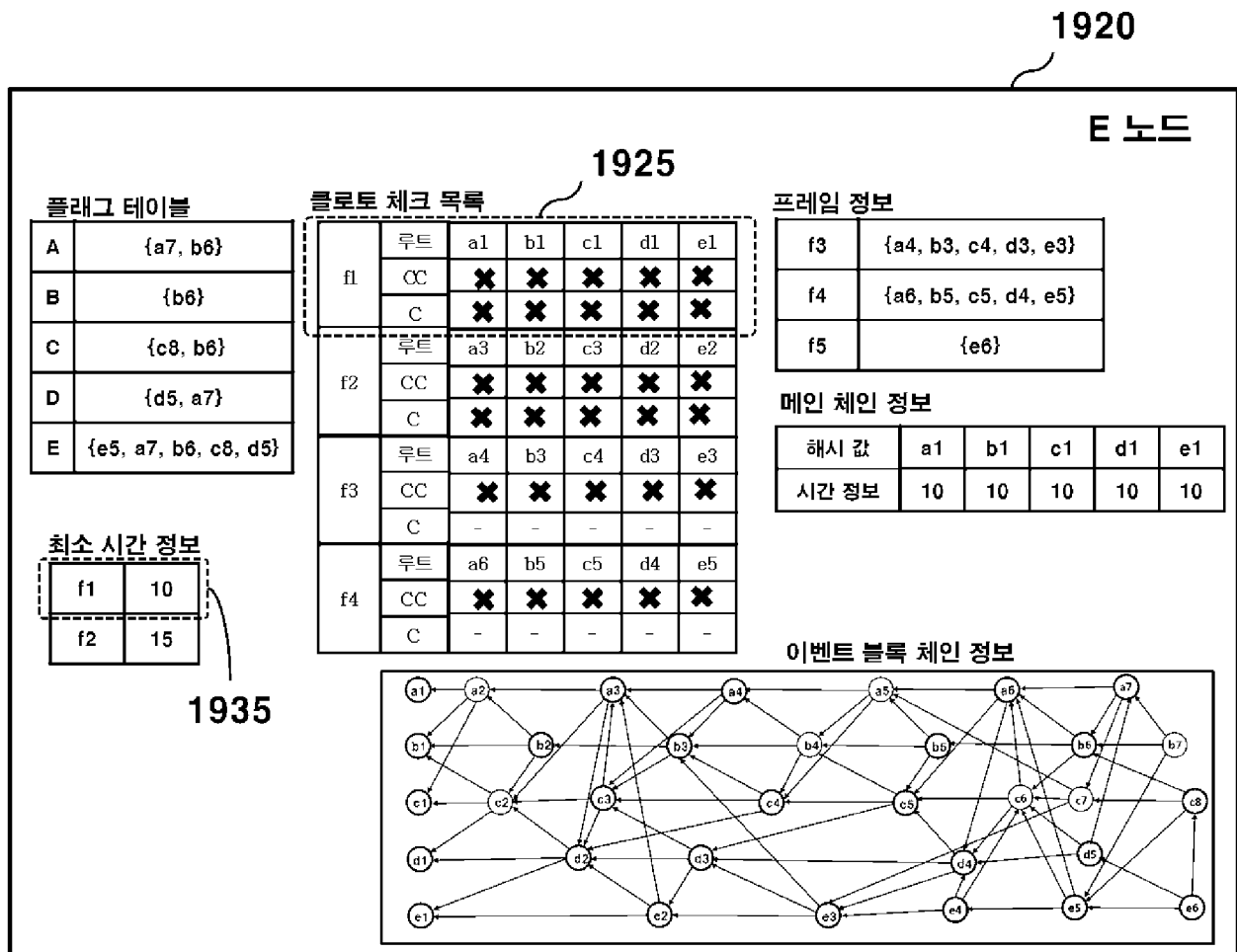
[도18d]



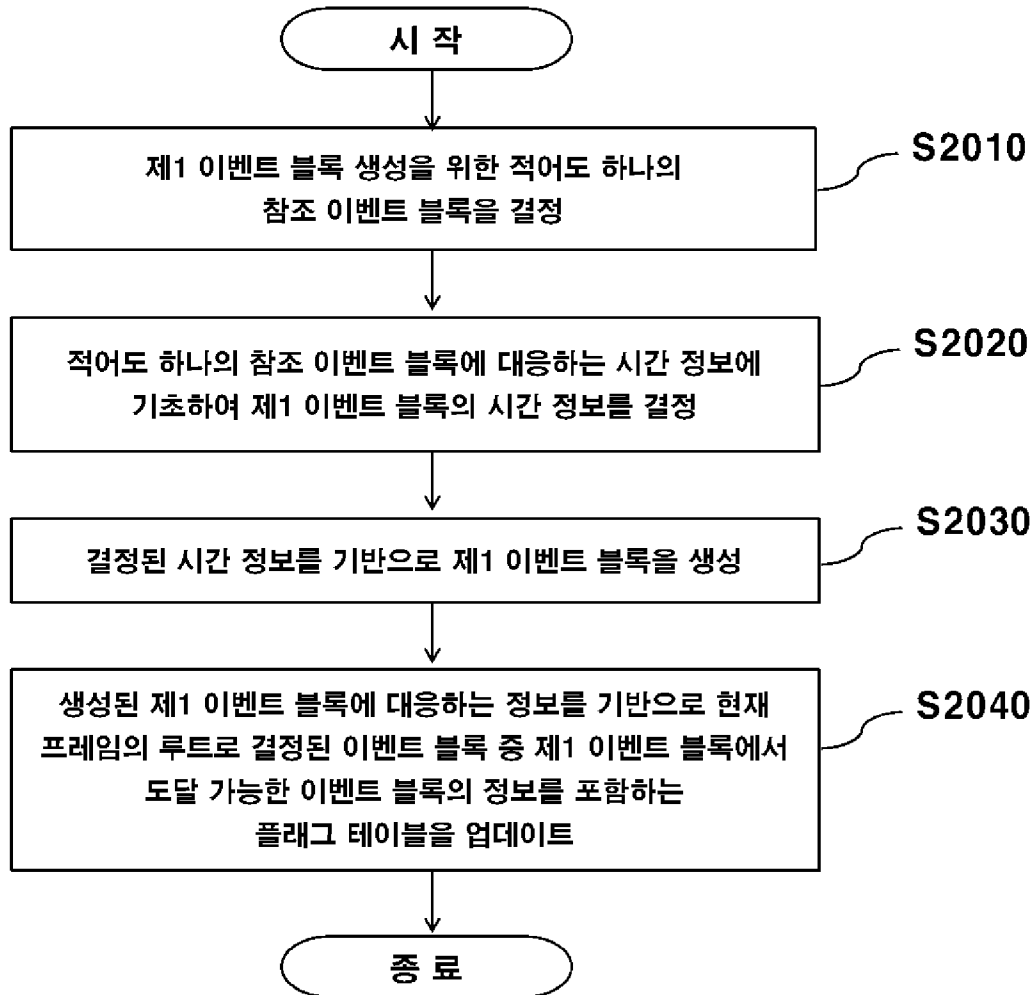
[도19a]



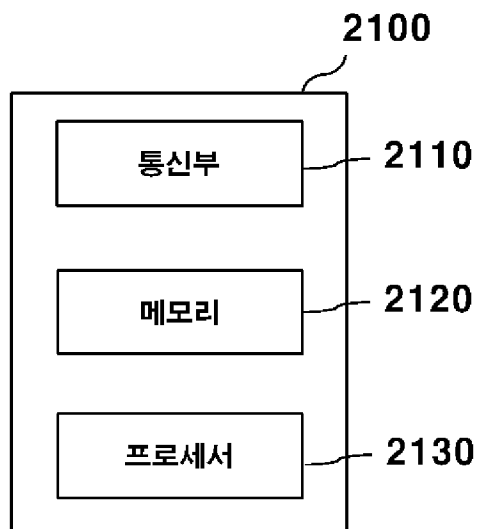
[도19b]



[도20]



[도21]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/KR2019/000425

A. CLASSIFICATION OF SUBJECT MATTER

H04L 29/06(2006.01)i, H04L 9/06(2006.01)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 29/06; H04L 9/06; H04L 9/32

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: blockchain, dispersion, time, reference, directed acyclic graph(DAG), agreement

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	SOMPOLINSKY, Yonatan et al. PHANTOM: A Scalable BlockDAG Protocol. Cryptology ePrint Archive: Report 2018/104. 29 May 2018 See pages 10-11.	1-15
A	SIIM, Janno. DAG-Based Distributed Ledgers. Research Seminar in Cryptology. 01 June 2018 See page 8.	1-15
A	TRAN, Alexandra. An Introduction to the blockDAG Paradigm. DAGlabs. 23 May 2018 See page 3.	1-15
A	SOMPOLINSKY, Yonatan et al. Inclusive Block Chain Protocols. Retrieved from <DOI: 10.1007/978-3-662-47854-7-33>. 22 February 2016 See section 2.	1-15
A	US 2017-0366353 A1 (ALTR SOLUTIONS, INC.) 21 December 2017 See claims 1-10 and figure 12.	1-15



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

“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

04 APRIL 2019 (04.04.2019)

Date of mailing of the international search report

05 APRIL 2019 (05.04.2019)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex Daejeon Building 4, 189, Cheongsu-ro, Seo-gu,
Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

Telephone No.

Information on patent family members

PCT/KR2019/000425

11/09/2018

A. 발명이 속하는 기술분류(국제특허분류(IPC))

H04L 29/06(2006.01)i, H04L 9/06(2006.01)i

B. 조사된 분야

조사된 최소문헌(국제특허분류를 기재)

H04L 29/06; H04L 9/06; H04L 9/32

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: 블록체인, 분산, 시간, 참조, directed acyclic graph(DAG), 합의

C. 관련 문헌

카테고리*	인용문헌명 및 관련 구절(해당하는 경우)의 기재	관련 청구항
A	YONATAN SOMPOLINSKY 등, 'PHANTOM: A Scalable BlockDAG Protocol', Cryptology ePrint Archive: Report 2018/104, 2018.05.29 페이지 10-11 참조.	1-15
A	JANNO SIIM, 'DAG-Based Distributed Ledgers', Research Seminar in Cryptology, 2018.06.01 페이지 8 참조.	1-15
A	ALEXANDRA TRAN, 'An Introduction to the blockDAG Paradigm', DAGlabs, 2018.05.23 페이지 3 참조.	1-15
A	YONATAN SOMPOLINSKY 등, 'Inclusive Block Chain Protocols', DOI: 10.1007/978-3-662-47854-7_33, 2016.02.22 섹션 2 참조.	1-15
A	US 2017-0366353 A1 (ALTR SOLUTIONS, INC.) 2017.12.21 청구항 1-10 및 도면 12 참조.	1-15

☐ 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

“T” 국제출원일 또는 우선일 후에 공개된 문헌으로, 출원과 상충하지 않으며 발명의 기초가 되는 원리나 이론을 이해하기 위해 인용된 문헌

“E” 국제출원일보다 빠른 출원일 또는 우선일을 가지나 국제출원일 이후에 공개된 선출원 또는 특허 문헌

“X” 특별한 관련이 있는 문헌. 해당 문헌 하나만으로 청구된 발명의 신규성 또는 진보성이 없는 것으로 본다.

“L” 우선권 주장에 의문을 제기하는 문헌 또는 다른 인용문헌의 공개일 또는 다른 특별한 이유(이유를 명시)를 밝히기 위하여 인용된 문헌

“Y” 특별한 관련이 있는 문헌. 해당 문헌이 하나 이상의 다른 문헌과 조합하는 경우로 그 조합이 당업자에게 자명한 경우 청구된 발명은 진보성이 없는 것으로 본다.

“O” 구두 개시, 사용, 전시 또는 기타 수단을 언급하고 있는 문헌

“&” 동일한 대응특허문헌에 속하는 문헌

“P” 우선일 이후에 공개되었으나 국제출원일 이전에 공개된 문헌

국제조사의 실제 완료일

2019년 04월 04일 (04.04.2019)

국제조사보고서 발송일

2019년 04월 05일 (05.04.2019)

ISA/KR의 명칭 및 우편주소



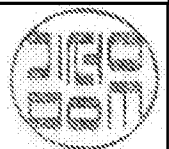
대한민국 특허청
(35208) 대전광역시 서구 청사로 189,
4동 (둔산동, 정부대전청사)

팩스 번호 +82-42-481-8578

심사관

김성우

전화번호 +82-42-481-3348



국제조사보고서에서
인용된 특허문헌

공개일

대응특허문헌

공개일

US 2017-0366353 A1

2017/12/21

US 10075298 B2

2018/09/11