



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2022년01월03일
(11) 등록번호 10-2346292
(24) 등록일자 2021년12월29일

(51) 국제특허분류(Int. Cl.)
G06F 9/50 (2018.01) G06F 16/18 (2019.01)
H04L 29/08 (2006.01)
(52) CPC특허분류
G06F 9/5066 (2013.01)
G06F 16/1865 (2019.01)
(21) 출원번호 10-2021-0053770
(22) 출원일자 2021년04월26일
심사청구일자 2021년04월26일
(30) 우선권주장
1020200149808 2020년11월11일 대한민국(KR)
(56) 선행기술조사문헌
KR1020200120863 A*
KR1020200080265 A
KR101986081 B1
KR1020200012705 A
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
엘에스웨어(주)
서울특별시 금천구 서부셋길 606, 18층 에이1801호외 (가산동, 대성디폴리스지식산업센터)
주식회사 퍼즐그룹
서울특별시 금천구 가산디지털1로 75-15, 1129호 (가산동)
(72) 발명자
김민수
경기도 성남시 분당구 내정로 151 305동 704호 (수내동, 양지마을금회 3단지 아파트)
신동명
서울 금천구 두산로 3가길 13 604호 대신드림빌 (뒷면에 계속)
(74) 대리인
윤재승

전체 청구항 수 : 총 8 항

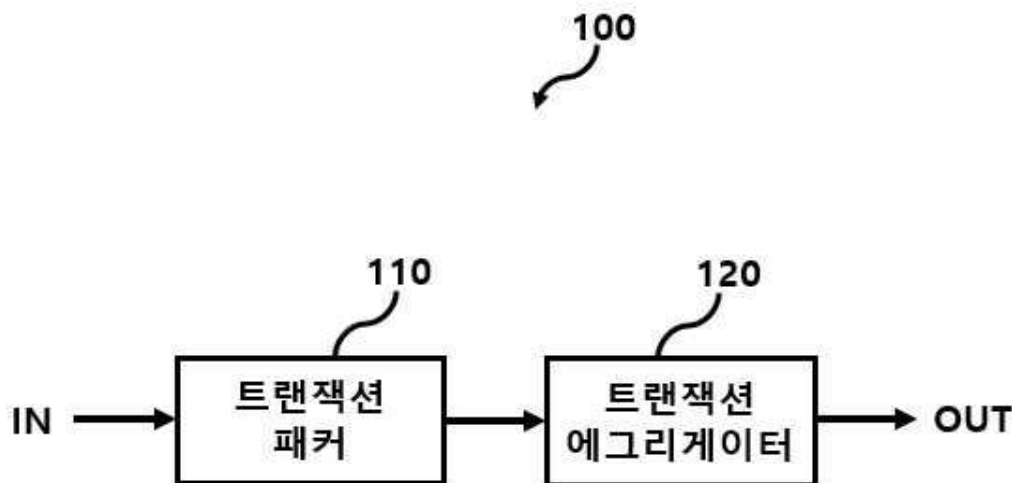
심사관 : 유진태

(54) 발명의 명칭 키 정보를 이용한 트랜잭션 분류 시스템 및 방법

(57) 요약

본 발명에 따른 키 정보를 이용한 트랜잭션 분류 시스템은, 사용자의 트랜잭션 요청에 따른 트랜잭션 정보를 나타내는 트랜잭션 프로포절(Tx Proposal)을 생성하고, 상기 생성된 트랜잭션 프로포절에 상기 트랜잭션 요청에 대응하는 키(key) 정보를 추가하는 트랜잭션 패커(Transaction Packer); 및 상기 트랜잭션 패커에서 전달받은 상기 트랜잭션 프로포절에 포함된 상기 키 정보에 따라 상기 트랜잭션 프로포절에 대응하는 공통 키셋 정보 및 공통 트랜잭션 배치(Transaction batch)정보를 생성하고, 상기 생성된 공통 트랜잭션 배치정보 별로 구분하여 미리 지정된 실행 노드그룹으로 전송하는 트랜잭션 에그리게이터(Transaction Aggregator)를 포함하는 것을 특징으로 한다.

대표도 - 도1



(52) CPC특허분류
G06F 9/5072 (2013.01)
H04L 67/1044 (2013.01)

조용준

서울특별시 금천구 가산디지털1로 75-15, 1129호
(가산동)

(72) 발명자

김성보

서울특별시 송파구 올림픽로32길 57-6 뉴파크플러
스 302호

이 발명을 지원한 국가연구개발사업

과제고유번호	1711116903
과제번호	2020-0-00063-001
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	블록체인융합기술개발(R&D)
연구과제명	소규모 노드환경에서 성능 극대화를 위한 멀티채널 기반 블록체인 기술 개발
기여율	1/1
과제수행기관명	엘에스웨어(주)
연구기간	2020.04.01 ~ 2020.12.31

명세서

청구범위

청구항 1

사용자의 트랜잭션 요청에 따른 트랜잭션 정보를 나타내는 트랜잭션 프로포절(Tx Proposal)을 생성하고, 상기 생성된 트랜잭션 프로포절에 상기 트랜잭션 요청에 대응하는 키(key) 정보를 추가하는 트랜잭션 패커(Transaction Packer); 및

상기 트랜잭션 패커에서 전달받은 상기 트랜잭션 프로포절에 포함된 상기 키 정보에 따라 상기 트랜잭션 프로포절에 대응하는 공통 키셋 정보 및 공통 트랜잭션 배치(Transaction batch)정보를 생성하고, 상기 생성된 공통 트랜잭션 배치정보 별로 구분하여 미리 지정된 실행 노드그룹으로 전송하는 트랜잭션 에그리게이터(Transaction Aggregator)를 포함하고,

상기 트랜잭션 패커는,

문자열 복합키의 일부에 해당하는 선두 문자열 또는 후미 문자열이 동일한 미확정 형태의 부분 키 정보를 상기 트랜잭션 프로포절에 추가하고,

상기 트랜잭션 에그리게이터는,

상기 부분 키 정보를 이용하여 상기 공통 키셋 정보 및 상기 공통 트랜잭션 배치정보를 생성하는 것을 특징으로 하는 키 정보를 이용한 트랜잭션 분류 시스템.

청구항 2

삭제

청구항 3

청구항 1에 있어서,

상기 공통 키셋 정보는,

복수의 트랜잭션 프로포절들에 포함되는 키 정보들 중 동일한 키를 갖는 것들에 대한 키 집합을 나타내는 것을 특징으로 하는 키 정보를 이용한 트랜잭션 분류 시스템.

청구항 4

청구항 1에 있어서,

상기 공통 트랜잭션 배치정보는,

상기 공통 키셋 정보에 포함된 키 정보들에 각각 대응하는 트랜잭션 프로포절들의 집합을 나타내는 것을 특징으로 하는 키 정보를 이용한 트랜잭션 분류 시스템.

청구항 5

청구항 1에 있어서,

상기 트랜잭션 에그리게이터는,

다수의 실행 노드들로부터 상기 실행 노드들 각각의 처리능력정보의 수신에 따라, 상기 공통 트랜잭션 배치정보의 계산을 위한 상기 실행 노드그룹을 지정하고, 상기 지정된 실행 노드그룹으로 상기 공통 트랜잭션 배치정보를 전송하는 것을 특징으로 하는 키 정보를 이용한 트랜잭션 분류 시스템.

청구항 6

트랜잭션 패커가 사용자의 트랜잭션 요청에 따른 트랜잭션 정보를 나타내는 트랜잭션 프로포절(Tx Proposal)을

생성하고, 상기 생성된 트랜잭션 프로포절에 상기 트랜잭션 요청에 대응하는 키(key) 정보를 추가하는 단계; 및 트랜잭션 에그리게이터가 상기 트랜잭션 프로포절에 포함된 상기 키 정보에 따라 상기 트랜잭션 프로포절에 대응하는 공통 키셋 정보 및 공통 트랜잭션 배치(Transaction batch)정보를 생성하고, 상기 생성된 공통 트랜잭션 배치정보 별로 구분하여 미리 지정된 실행 노드그룹으로 전송하는 단계를 포함하고,

상기 트랜잭션 패커는,

문자열 복합키의 일부에 해당하는 선두 문자열 또는 후미 문자열이 동일한 미확정 형태의 부분 키 정보를 상기 트랜잭션 프로포절에 추가하고,

상기 트랜잭션 에그리게이터는,

상기 부분 키 정보를 이용하여 상기 공통 키셋 정보 및 상기 공통 트랜잭션 배치정보를 생성하는 것을 특징으로 하는 키 정보를 이용한 트랜잭션 분류방법.

청구항 7

삭제

청구항 8

청구항 6에 있어서,

상기 공통 키셋 정보는,

복수의 트랜잭션 프로포절들에 포함되는 키 정보들 중 동일한 키를 갖는 것들에 대한 키 집합을 나타내는 것을 특징으로 하는 키 정보를 이용한 트랜잭션 분류방법.

청구항 9

청구항 6에 있어서,

상기 공통 트랜잭션 배치정보는,

상기 공통 키셋 정보에 포함된 키 정보들에 각각 대응하는 트랜잭션 프로포절들의 집합을 나타내는 것을 특징으로 하는 키 정보를 이용한 트랜잭션 분류방법.

청구항 10

청구항 6에 있어서,

상기 생성된 공통 트랜잭션 배치정보 별로 구분하여 미리 지정된 실행 노드그룹으로 전송하는 단계는,

다수의 실행 노드들로부터 상기 실행 노드들 각각의 처리능력정보의 수신에 따라, 상기 공통 트랜잭션 배치정보의 계산을 위한 상기 실행 노드그룹을 지정하고, 상기 지정된 실행 노드그룹으로 상기 공통 트랜잭션 배치정보를 전송하는 것을 특징으로 하는 키 정보를 이용한 트랜잭션 분류방법.

발명의 설명

기술 분야

[0001] 본 발명은 블록체인 프레임워크에서 효율적인 트랜잭션 분산처리를 위해, 트랜잭션 요청에 대응하는 공통 키셋 정보를 이용하여 분산처리가 이루어질 수 있도록 하는 키 정보를 이용한 트랜잭션 분류 기술에 관한 것이다.

배경 기술

[0002] 병행 컴퓨팅의 큰 문제 중 하나는, 하나의 변수에 대한 참조 및 변경에 의한 충돌이다. 가장 일반적인 방법은 단일 변수에 대한 배타적 접근 제어를 위한 Lock Strategy이지만, 1993년 이후 STM(Software Transactional Memory)이나 RLU (Read-Log-Update) 등 Lock-Free Strategy가 개발되었다. 대표적인 Lock-Free Strategy인 STM은 SMP (Shared Memory Multi-Processor) 같은 단일 머신 내에서 완결되는 병렬처리, etcd와 같은 분산 컴퓨팅

상의 메모리 공유 모델, 그리고 블록체인의 분산 처리된 트랜잭션 간 충돌 검출 등에서 활용하는 병렬처리 기법이다.

[0003] 블록체인 기술은 관리 주체를 분산시킬 수 있다는 그 독특한 특성으로 인해, 기존 산업에 적용되면 그 파급효과는 매우 클 것으로 예상되어왔다. 하지만, 기존 산업에 대한 블록체인 기술의 적용은 예상보다 늦어지고 있는데, 그 이유 중 하나는 처리 성능(throughput)이 매우 낮다는 점이다. 이러한 블록체인의 성능 문제에 관하여 합의 알고리즘과 멀티체인과 같은 기술을 통해 합의 시간 감소 및 블록 당 처리 용량 증가 및 체인 당 트랜잭션 분산을 통한 플랫폼의 전체 처리 성능 향상을 피하고 있다.

[0004] 일반적으로 마이닝이란 1. 실행할 트랜잭션 선택, 2. 트랜잭션 실행(Execution), 3. 트랜잭션 기록 순서 결정(Ordering), 4. Hash & Nonce Finding의 총 4가지 작업을 통칭한다.

[0005] 일반적으로 합의 알고리즘 개발은, 위의 3번째 및 4번째 단계 및 검증 과정의 최적화 및 고속화에 그 방점을 둔다. 초창기의 블록체인 프레임워크는 Public 블록체인을 위한 비트코인의 PoW 합의 방식을 답습하였기 때문에, 3번째 및 4번째 단계의 속도 향상을 위해서 안전하면서 효율적인 합의 알고리즘이 개발되어왔다. 하지만, 기존의 합의 알고리즘에 관한 연구는 2번째 단계인 트랜잭션 실행의 고속화는 그 대상이 아니다. 합의 알고리즘의 개선과는 다른 접근 방식으로, 체인별로 서로 다른 영역 또는 유형의 트랜잭션 요청을 처리함으로써 전체 처리 성능을 향상하는 멀티 체인(멀티채널, 사이드 체인) 방식도 있다. 블록을 체인마다 별도로 처리 및 생성함으로써 전체 성능(throughput)을 향상할 수 있으나, 체인 간에 공유하는 정보 혹은 체인 간의 상호 연동을 위한 변수와 같이 서로 다른 체인이 같은 키를 참조 및 변경하고자 하는 경우, 단일 블록 키 충돌 문제로 인한 성능 저하 문제가 발생한다. 이는 재고 관리와 같은 기존 산업이 요구하는 프로세스를 블록체인 상에 그대로 구현하는 경우, 효율적으로 처리하는 것이 어렵다는 것을 의미한다.

[0006] 블록체인은 상호 검증을 위해 분산 노드로 구성되어 있으나, 노드 수에 따른 성능 향상(Scalability)이 불가능하다는 고질적인 문제를 가지고 있다. 블록체인에서 Scalability를 달성하기 어려운 이유 중 가장 대표적인 요인은 단일 블록 키 충돌 문제이다.

[0007] 단일 블록 키 충돌 문제는 둘 이상의 계산 주체(트랜잭션 요청을 실행하는 주체)에서 실행되는 트랜잭션이 같은 키를 참조 및 변경하는 경우, 해당 트랜잭션 간에 그 키가 충돌하기 때문에 한 블록에서 한 번에 처리할 수 없다. 이는 같은 키를 참조하는 트랜잭션이 참조 후 변경한 값이 다르면(R/W Set이 충돌하면) 원장 기록의 일관성이 깨지기 때문이다. 이는 STM의 R/W Set validation에 해당하는 과정으로, 블록체인에서도 STM과 같이 트랜잭션을 반영할 때 R/W Set이 충돌하는 경우, 해당 트랜잭션의 처리 결과를 파기하고 재실행 후, 다음 블록에서 반영을 시도해야 한다. 결과적으로 상호 간에 간섭하는 트랜잭션을 확정하기 위해서는 간섭하는 트랜잭션의 개수만큼의 블록이 필요하다.

[0008] 상호 간에 변경된 값을 공유할 수 있다면 이러한 문제를 해결할 수 있으나, 블록체인은 분산 노드로 구성되어있기 때문에 공유 메모리가 아닌 분산 공유 메모리 모델을 사용해야 하지만, 이는 더 큰 성능 저하를 의미하기 때문에, 서로 다른 노드 간에 처리 결과를 공유하는 것은 바람직하지 않다.

[0009] 결과적으로, 단일 블록 키 충돌 문제는 블록체인의 성능 저하를 의미할 뿐만이 아니라, 블록 밀도 감소(한 블록에 담기는 트랜잭션 수 감소), 트랜잭션 계산자원 효율 저하(동일 트랜잭션을 반복해서 재실행함으로써, 유효 트랜잭션 실행률 저하) 등의 문제를 발생시킨다. 선행기술문헌 : 한국공개특허 제10-2018-0115778호

발명의 내용

해결하려는 과제

[0010] 본 발명이 해결하고자 하는 과제는 단일 원장 블록체인 프레임워크에서 효율적인 트랜잭션 분산처리를 위해, 공통 키셋정보를 이용한 트랜잭션 분산처리를 위한 트랜잭션 분류 시스템 및 방법에 관한 것이다.

과제의 해결 수단

[0011] 상기의 과제를 해결하기 위한 본 발명에 따른 키 정보를 이용한 트랜잭션 분류 시스템은, 사용자의 트랜잭션 요청에 따른 트랜잭션 정보를 나타내는 트랜잭션 프로포절(Transaction Proposal)을 생성하고, 상기 생성된 트랜잭션 프로포절에 상기 트랜잭션 요청에 대응하는 키(key) 정보를 추가하는 트랜잭션 패커(Transaction Packer); 및 상기 트랜잭션 패커에서 전달받은 상기 트랜잭션 프로포절에 포함된 상기 키 정보에 따라 상기 트랜잭션 프로포절에 대응하는 공통 키셋 정보 및 공통 트랜잭션 배치(Transaction batch)정보를 생성하고, 상기 생성된 공

통 트랜잭션 배치정보 별로 구분하여 미리 지정된 실행 노드그룹으로 전송하는 트랜잭션 에그리게이터 (Transaction Aggregator)를 포함하는 것을 특징으로 한다.

- [0012] 상기 키 정보는, 배열이나 복합키를 포함하는 미확정 형태의 부분 키 정보를 포함하는 것을 특징으로 한다.
- [0013] 상기 공통 키셋 정보는, 복수의 트랜잭션 프로포절들에 포함되는 키 정보들 중 동일한 키를 갖는 것들에 대한 키 집합을 나타내는 것을 특징으로 한다.
- [0014] 상기 공통 트랜잭션 배치정보는, 상기 공통 키셋 정보에 포함된 키 정보들에 각각 대응하는 트랜잭션 프로포절들의 집합을 나타내는 것을 특징으로 한다.
- [0015] 상기 트랜잭션 에그리게이터는, 다수의 실행 노드들로부터 상기 실행 노드들 각각의 처리능력정보의 수신에 따라, 상기 공통 트랜잭션 배치정보의 계산을 위한 상기 실행 노드그룹을 지정하고, 상기 지정된 실행 노드그룹으로 상기 공통 트랜잭션 배치정보를 전송하는 것을 특징으로 한다.
- [0017] 상기의 과제를 해결하기 위한 본 발명에 따른 키 정보를 이용한 트랜잭션 분류방법은, 사용자의 트랜잭션 요청에 따른 트랜잭션 정보를 나타내는 트랜잭션 프로포절(Transaction Proposal)을 생성하고, 상기 생성된 트랜잭션 프로포절에 상기 트랜잭션 요청에 대응하는 키(key) 정보를 추가하는 단계; 및 상기 트랜잭션 패커에서 전달 받은 상기 트랜잭션 프로포절에 포함된 상기 키 정보에 따라 상기 트랜잭션 프로포절에 대응하는 공통 키셋 정보 및 공통 트랜잭션 배치(Transaction batch)정보를 생성하고, 상기 생성된 공통 트랜잭션 배치정보 별로 구분하여 미리 지정된 실행 노드그룹으로 전송하는 단계를 포함하는 것을 특징으로 한다.
- [0018] 상기 키 정보는, 배열이나 복합키를 포함하는 미확정 형태의 부분 키 정보를 포함하는 것을 특징으로 한다.
- [0019] 상기 공통 키셋 정보는, 복수의 트랜잭션 프로포절들에 포함되는 키 정보들 중 동일한 키를 갖는 것들에 대한 키 집합을 나타내는 것을 특징으로 한다.
- [0020] 상기 공통 트랜잭션 배치정보는, 상기 공통 키셋 정보에 포함된 키 정보들에 각각 대응하는 트랜잭션 프로포절들의 집합을 나타내는 것을 특징으로 한다.
- [0021] 상기 생성된 공통 트랜잭션 배치정보 별로 구분하여 미리 지정된 실행 노드그룹으로 전송하는 단계는, 다수의 실행 노드들로부터 상기 실행 노드들 각각의 처리능력정보의 수신에 따라, 상기 공통 트랜잭션 배치정보의 계산을 위한 상기 실행 노드그룹을 지정하고, 상기 지정된 실행 노드그룹으로 상기 공통 트랜잭션 배치정보를 전송하는 것을 특징으로 한다.

발명의 효과

- [0022] 본 발명에 따르면, 복수의 트랜잭션 요청들에 대해, 이를 처리할 실행 노드들에 상호 간섭이 없도록 분류하여 실행할 수 있도록 공통 키셋정보를 제공함으로써, 기존의 멀티체인 블록체인 기술 및 Transaction Pre-Ordering 기술과 달리 높은 트랜잭션 유효 실행 효율과 Scalability를 동시에 달성할 수 있도록 한다.
- [0023] 예를 들어, 단일 블록 키 충돌이 없는 실행 결과를 보장하는 노드 그룹별 트랜잭션 그룹을 충분히 빠른 속도로 생성할 수 있다. 공통 트랜잭션 배치정보를 블록체인 상의 노드 그룹에 분할 배분하여, 이를 순차적 또는 병행적으로 처리함으로써 노드 수 증가에 따른 Scalability를 얻을 수 있는 블록체인 프레임워크 기술을 구현할 수 있다.

도면의 간단한 설명

- [0024] 도 1은 본 발명에 따른 키 정보를 이용한 트랜잭션 분류 시스템을 설명하기 위한 일 실시예의 구성 블록도이다.
- 도 2는 사용자의 요청에 따른 트랜잭션들을 예시하는 참조도이다.
- 도 3은 트랜잭션 에그리게이터가 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 일 예의 참조도이다.
- 도 4는 트랜잭션 에그리게이터가 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 다른 예의 참조도이다.
- 도 5는 트랜잭션 에그리게이터가 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 또다른 예의 참조도이다.

도 6은 트랜잭션 에그리게이터가 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 또다른 예의 참조도이다.

도 7은 트랜잭션 에그리게이터가 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 또다른 예의 참조도이다.

도 8은 트랜잭션 에그리게이터가 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 또다른 예의 참조도이다.

도 9는 트랜잭션 에그리게이터가 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 또다른 예의 참조도이다.

도 10은 트랜잭션 에그리게이터가 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 또다른 예의 참조도이다.

도 11은 트랜잭션 에그리게이터가 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 또다른 예의 참조도이다.

도 12는 트랜잭션 에그리게이터가 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 또다른 예의 참조도이다.

도 13은 트랜잭션 에그리게이터가 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 또다른 예의 참조도이다.

도 14는 본 발명에 따른 키 정보를 이용한 트랜잭션 분류방법을 설명하기 위한 일 실시예의 플로차트이다.

발명을 실시하기 위한 구체적인 내용

- [0025] 이하, 첨부된 도면을 참조하여 본 발명의 바람직한 실시예를 상세히 설명하기로 한다.
- [0026] 본 발명의 실시예들은 당해 기술 분야에서 통상의 지식을 가진 자에게 본 발명을 더욱 완전하게 설명하기 위하여 제공되는 것이며, 아래의 실시예들은 여러 가지 다른 형태로 변형될 수 있으며, 본 발명의 범위가 아래의 실시예들로 한정되는 것은 아니다. 오히려, 이들 실시예는 본 개시를 더욱 충실하고 완전하게 하며 당업자에게 본 발명의 사상을 완전하게 전달하기 위하여 제공되는 것이다.
- [0027] 본 명세서에서 사용된 용어는 특정 실시예를 설명하기 위하여 사용되며, 본 발명을 제한하기 위한 것이 아니다. 본 명세서에서 사용된 바와 같이 단수 형태는 문맥상 다른 경우를 분명히 지적하는 것이 아니라면, 복수의 형태를 포함할 수 있다. 또한, 본 명세서에서 사용된 바와 같이, 용어 "및/또는"은 해당 열거된 항목 중 어느 하나 및 하나 이상의 모든 조합을 포함한다.
- [0029] 본 발명이 제안하는 기술은 트랜잭션 병렬처리 기술을 블록체인 프레임워크에 응용한 것으로, 서로 다른 노드에서 처리되는 트랜잭션 간에 단일 블록 키 충돌 문제가 발생하지 않도록 트랜잭션 요청을 나눔으로써, 블록체인의 병렬처리 효율을 높이기 위한 방식이다.
- [0030] 이 방식은 크게, 트랜잭션 요청에 따른 키 정보 생성, 생성된 키 정보에 기초하여 공통의 키셋 정보를 갖는 트랜잭션들끼리 분류(Transaction Proposal Aggregation: TPA) 및 분류된 트랜잭션들끼리 계산을 실행할 수 있도록 하는 Cached Tx Executor으로 구성된다. TPA는 실시간으로 주어지는 트랜잭션 요청을 그룹 간에 키 중복이 없도록 노드 그룹별로 분류함으로써, Cached Tx Executor는 분류·할당된 트랜잭션을 순서대로 실행한다.
- [0031] 먼저, 본발명 관련한 용어 및 데이터 구조 등에 대한 상세한 내용을 정리하면 다음과 같다.
- [0032] Tx: 트랜잭션의 약자이다. Tx는 하나의 처리를 의미하며, 적용 대상의 블록체인 아키텍처가 요구하는 트랜잭션 처리에 필요한 정보로 구성된다.
- [0033] Tx Proposal: 트랜잭션 요청에 따른 트랜잭션 정보를 포함한다.
- [0034] Tx Simulation: 트랜잭션 요청에 대한 트랜잭션 실행기에서의 실행 결과를 의미한다.
- [0035] Ledger Key: 원장에 기록되는 값의 키(변수명)을 의미한다. 원장에 기록되지 않는 변수 또는 구조체 내의 field 명 등을 의미하지 않는다.
- [0036] Partial Ledger Key: 원장에 기록되는 값의 키의 일부로 구성되는 정보이다.

- [0037] R/W Key Set (RWKS): 임의의 Tx Proposal의 실행시, 참조 및 변경한 값의 Ledger Key 정보의 집합을 의미한다.
- [0038] eTP (encapsulated Tx Proposal): 기존의 Tx Proposal에 트랜잭션 에그리게이터에 전달하기 위한 R/W Key Set Description 정보가 추가된 Tx Proposal이다.
- [0039] Tx Tree는 그 자체가 하나의 Tx 유형의 분류를 표현하는 표현형인 동시에, 분류된 Tx의 집합을 표현하는 데이터 구조이다.
- [0040] Tx Tree는 크게 2가지의 데이터로 구성된다. 하나는 Tx Tree의 RWKS으로, 해당 Tx Tree의 유형 분류 기준을 표현하는 정보이다. 또 하나는 Tx Set으로, 해당 Tx Tree의 유형 분류 기준에 부합하는 Tx의 집합이다.
- [0041] eTPB: 상호 영향이 있는 eTP의 순서가 정해진 묶음이다. eTPB내의 순서는 eTPID(TxID)에 의해 정의된다.
- [0042] eTPBs(Forest, ForestContainer): 순서가 정해진 eTPB의 묶음이다. eTPBs내의 eTPB의 순서는eTPBID(TreeID)에 의해 정의된다.
- [0043] eTSBs: eTPBs에서 정해진 순서대로 트랜잭션 실행기에서 실행된 Tx Simulation의 묶음이다.
- [0044] eTPRBs: eTPBs에서 정해진 순서대로 실행된 Tx Proposal Response의 묶음이다.
- [0045] NodeInfoMap: 블록체인에 참가한 모든 Node의 상태 정보를 의미한다.
- [0046] Capacity: 각 노드 그룹이 한 블록 생성 주기 내에 처리할 수 있는 Tx 처리 성능 용량을 의미한다.
- [0047] Arithmetic Load: 원장에 대한 키 참조 및 변경 조작을 포함하지 않는 Tx 실행에 걸리는 시간을 의미한다.
- [0048] Key load: 원장에 대한 키 참조 및 변경에 필요한 실행 시간을 의미한다.
- [0050] eTP는 다음과 같은 정보로 구성된다.

표 1

종류	내용
EtpId (string)	eTP의 identifier. 일반적으로 Tx Proposal의 Tx ID를 준용함
TaskLoad (int32)	해당 Tx Proposal을 실행하는데 필요한 cost(=시간). 단 원장에서 값을 읽거나 쓰는데 걸리는 시간을 제외함
ParKeys ([]string, 또는 [][]string))	해당 Tx Proposal이 참조·변경할 가능성이 있는 Ledger Key의 부분키(Partial Ledger Key)의 배열
Keys ([]string)	해당 Tx Proposal이 참조·변경할 가능성이 있는 Ledger Key의 배열
Proposal (적용 블록체인 플랫폼에 따라 바뀜)	적용 블록체인 플랫폼의 Tx Proposal의 원 데이터(Tx Proposal을 그대로 넣으면 됨)
Receptor (string)	해당 Tx Proposal을 접수한 Node(API Server)의 정보. 본 문서에서는 다루지 않음.

[0052] eTPB는 다음과 같은 정보로 구성된다.

표 2

종류	내용
TreeId (string)	해당 Tree의 identifier
TreeCoreLoad (int32)	해당 Tree에 담긴 Tx Proposal의 TaskLoad의 합
ParKeys (map[string]bool, 또는 map[][]string]bool)	해당 Tree에 담긴 모든 Tx Proposal이 참조·변경할 가능성이 있는 Ledger Key의 부분키(Partial Ledger Key)의 배열
Keys (map[string]bool)	해당 Tree에 담긴 모든 Tx Proposal이 참조·변경할 가능성이 있는 Ledger Key의 배열
Etps ([]ETP)	해당 Tree에 담긴 Tx Proposal의 배열

[0054] eTPBs는 다음과 같은 정보로 구성된다.

표 3

종류	내용
Forest ([]eTPB)	전송 대상의 Node Group에서 처리하는 모든 eTP Tree의 집합 eTP Tree의 데이터 적재 순서는 보장되지 않으므로, eTP Tree의 TreeID에 의해 순서가 결정됨
Forest ID (NodeGroupID:int32)	해당 eTPB의 전송 대상 Node Group ID
ForestLoad (int32)	해당 eTPBs를 전부 실행할 때의 부하값

[0057] dTPA(distributed TPA)간 통신 프로토콜

[0058] dTPA는 네트워크 상에 분산되어 존재하는 TPA 머신이다. dTPA는 각 API Server로부터 독립적으로 또는 동시에 eTP 정보를 입력받는다.

[0059] 각 dTPA는 동일한 알고리즘을 사용하여 eTP를 분류하나, 복수의 eTP 정보를 입력받는 순서 및 방법에 따라서 모든 TPA에서 동일한 방식으로 eTP를 분류할 수 없다. 이 문제를 해결하기 위해서 dTPA는 상호간에 정보를 선택적 또는 비선택적으로 공유함으로써 모든 TPA가 동일한 방식으로 eTP를 분류하고, 그 분류 결과를 각 Endorsing Peer 노드에 대해 중복되지 않게 전송할 수 있어야 한다.

[0060] 이를 위해서 dTPA는 다음과 같은 정보를 상호간에 교환한다.

표 4

종류	통신 방식	내용
eTP_Charge	Broadcast	입의의 eTP를 특정 TPA에서 담당하겠다는 신호 eTP 및 TPA_Id로 구성됨
Charge_Negotiate	P2P	입의의 eTP에 대한 담당 TPA 선언이 중복되는 경우, 해결하기 위한 통신 Negotiation 알고리즘에 따라서 데이터 구성은 달라질 수 있음
eTPBs_Broadcast	Broadcast	발신 시점에서 TPA가 현재까지 담당하고 있는 eTPBs를 다른 TPA에 알리기 위한 신호 eTPBs 또는 Δ eTPBs 정보를 공유
RWKS_Broadcast	Broadcast	발신 시점에서 TPA가 현재까지 담당하고 있는 eTPBs의 RWKS를 다른 TPA에 알리기 위한 신호 eTPBs 또는 Δ eTPBs 정보를 공유
eTP_Toss	P2P	해당 TPA가 담당할 수 없는 eTP를 대상 TPA에 넘기기 위한 통신
eTPB_Exchange_Negotiation	P2P	TPA간에 eTPB를 상호 교환하기 위한 Negotiation 신호
eTPB_Exchange	P2P	TPA간에 eTPB를 상호 교환하기 위한 Negotiation 신호

[0062] 특정 eTP에 대한 담당 선언이 중복되는 경우, Charge Negotiation을 한 TPA들은 그 결정 결과를 다른 TPA에 전달할 필요가 없다. 선언을 하지 않은 TPA는 그 eTP에 대한 분류 작업을 하지 않고, 그 결과만 eTPB_Broadcast를 통해서 전달받으면 되기 때문이다.혹여, 아직 그 분류 결과를 전달받지 못한 시점에 해당 eTP와 변수 중복이 있는 eTP를 전달받은 경우, 해당 TPA는 그 eTP에 대해서 담당 선언을 했던 모든 TPA에게 그 eTP를 eTP_Toss를 통해 전달하면 된다. dTPA의 데이터 구조는 다음과 같다.

표 5

종류	데이터구조
eTP_Charge	eTP(ETP), TPA_Id(int32)
Charge_Negotiate	예) TPA_Id(int32), related_tx_count(int32), related_forest_count(int32)
eTPBs_Broadcast	eTPBs 전체, 또는 변경된 eTPBs 이력인 Δ eTPBs에 포함된 eTP의 ID

RWKS_Broadcast	eTPBs 전체, 또는 변경된 eTPBs 이력인 ΔeTPBs에 포함된 eTPBs의 Ledger Key RWKS 및 Partial Key RWKS
eTP_Toss	eTP 전체
eTPB_Exchange_Negotiation	예) eTPB_IDs([int32), eTPB_IDs([int32), bool (è 주고자 하는 eTPB의 ID, 받고자 하는 eTPB의 ID, 승낙 여부)
eTPB_Exchange	[]eTPB

- [0065] 이하, 본 발명에 따른 키 정보를 이용한 트랜잭션 분류 시스템 및 방법을 상세히 설명한다.
- [0066] 도 1은 본 발명에 따른 키 정보를 이용한 트랜잭션 분류 시스템(이하, 블록체인 시스템이라 칭함)을 설명하기 위한 일 실시예의 구성 블록도이다.
- [0067] 도 1을 참조하면, 블록체인 시스템(100)은 트랜잭션 패커(Transaction Packer: 110), 트랜잭션 에그리게이터(Transaction Aggregator: 120)를 포함한다.
- [0068] 트랜잭션 패커(110)는 사용자의 트랜잭션 요청(IN)에 따른 트랜잭션 정보를 나타내는 트랜잭션 프로포절(Tx Proposal)을 생성하고, 상기 생성된 트랜잭션 프로포절에 상기 트랜잭션 요청에 대응하는 키(key) 정보를 추가하여, 트랜잭션 에그리게이터(Transaction Aggregator: 120)로 전달한다.
- [0069] 트랜잭션 프로포절은 사용자의 트랜잭션 요청에 대응하는 트랜잭션 정보를 포함하는 것으로, 트랜잭션 에그리게이터(120)에서 사용될 수 있는 트랜잭션의 데이터 형식을 나타낸다.
- [0070] 키(key) 정보는 전술한 용어 정리에서 Ledger Key를 의미한다. 또한, 키(key) 정보는 배열이나 복합키를 포함하는 미확정 형태의 부분 키 정보를 포함할 수 있다. 이는 전술한 용어 정리에서, Partial Ledger Key를 의미한다.
- [0071] 다양한 케이스에서 참조 및 변경하는 Ledger Key는 그 Tx를 실행하는 시점에 결정되거나, Tx를 실행해보기 전에 전부 파악하는 것이 어려운 경우가 있다. Partial Ledger Key는 다음과 같은 상황에서 활용한다.
- [0072] 해당 Tx에서 참조 변경하는 Ledger Key의 부분집합이 어떠한 특징을 공유하는 경우이다. 배열인 경우(배열 abc의 모든 또는 일부의 값에 접근하는 경우), 선두 문자열이 동일한 경우, 후미 문자열이 동일한 경우, 일부 문자열을 포함하는 경우 등이 있을 수 있다. 또한, 해당 Tx에서 참조 및 변경하는 Ledger Key가 실행 도중에 생성 또는 취득되며, 그 특징을 사전에 알고 있는 경우이다.
- [0073] 배열은 선두 문자열의 경우와 동일하게 취급 가능하고, 후미 문자열도 문자열을 반전시키면 동일하게 사용 가능하고, 일부 문자열 일치 방식 또한 데이터 구조 및 알고리즘을 적절하게 수정하면 사용 가능하다. 단, 후미 문자열 일치 및 일부 문자열 일치 추가하는 경우에는 Dictionary Layer를 하나씩 더 추가하면 된다. 설명을 위해서, 선두 문자열이 동일한 경우에 대해서 다룬다.
- [0074] R/W Key Set (RWKS)을 Ledger Key에 대해서 뿐만이 아니라 Partial Ledger Key와 Ledger Key 두가지를 위해서 준비한다. 이를 테면, Partial Ledger Key 및 Ledger Key를 보관하는 Dictionary의 데이터 구조로서 Trie를 사용할 수 있다. 어떤 Smart Contract의 Tx에서 주로 “APPLESEED”, “APPLEJUICE”와 같이 “APPLE”로 시작하는 문자열에 접근하는 것을 알 수 있는 경우, 이 Tx의 Partial Key는 “APPLE*”으로 표현된다.
- [0075] 이렇게, Partial Ledger Key가 주어지는 경우, 새로 주어진 Tx A의 Partial Ledger Key RWKS의 모든 Partial Ledger Key K를 가지고, 모든Tx Tree B₁~B_k의 Partial Ledger Key에 대한 RWKS에서 K를 검색한다. 이러한 알고리즘과 동일한 내용을 Partial Ledger Key K와 Partial Ledger Key의 RWKS에 대해서 수행한다.
- [0076] Partial Ledger Key K를 발견하지 못한 경우, 위의 알고리즘을 그대로 실행함으로써, Partial Ledger Key를 포함하는 Tx에 대해서도 위의 알고리즘을 통해 Tx를 분류할 수 있다.
- [0077] 한편, 위의 Partial Key Tx Aggregation으로도 분류가 어려운 Tx가 존재한다. 해당 Tx에서 참조 및 변경하는 Ledger Key가 실행 도중에 생성 또는 취득되며, 그 특징을 사전에 알 수 없는 경우이다. Random하게 정해지는 Ledger Key의 경우, 원장상의 값을 Ledger Key로 삼는 경우 등이다. 이와 같은 경우에는, Uncertain Key를 포함하는 Tx를 하나의 Tx Tree로 분류하고, 분류된 Uncertain Key Tx Tree의 크기가 큰 경우, 적절한 크기까지만 채택하고, 나머지는 보류한다. 위의 알고리즘으로 분류된 모든 eTPBs에 Uncertain Key Tx Tree를 추가한다.
- [0078] 트랜잭션 실행기에서 이 Uncertain Key Tx Tree를 가장 먼저 실행한 다음에 그 실행 결과(Cache)상에서 나머지

Tx Tree를 실행하면 Uncertain Key로 인해 분류하지 못한 Tx들에 대해서도 Tx Tree간 R/W Set의 충돌 없이 병렬 처리하는 것이 가능하다. 이를테면, 해당 Smart Contract에 대한 argument로 주어진 값이 선두 문자열로 활용되는 경우에는, 그 값 “XYZ”를 사용해서 “XYZ*”와 같이 생성한다. 여기서 ‘*’ 문자는 일반 문자가 아닌 특수 문자로 생각한다.

- [0079] 트랜잭션 패커(110)는 사용자의 트랜잭션 요청을 Tx Proposal으로 변환하고, 부가 정보를 추가하여 eTP(encapsulated Tx Proposal)를 생성하고, 생성된 eTP를 트랜잭션 에그리게이터(120)에 전달한다. 여기서, eTP는 기존의 Tx Proposal에 트랜잭션 에그리게이터(120)에 전달하기 위해 부가정보(R/W Key Set Description 정보)가 추가된 Tx Proposal이다.
- [0080] 도 2는 사용자의 요청에 따른 트랜잭션들(Tx 1 ~ Tx 11)을 예시하는 참조도이다. 도 2를 참조하면, 트랜잭션들(Tx 1 ~ Tx 11)에서 A, B, C, D, E, F, G, H는 금전 거래에 있어서 송금자와 수금자를 예시하는 것일 수 있다. 예를 들어, Tx 1: A -> C는 A라는 사람이 C라는 사람에게 일정 금액을 송금하는 트랜잭션을 예시하는 것일 수 있다.
- [0081] 트랜잭션 패커(110)는 사용자들의 트랜잭션 요청에 따라, 트랜잭션들(Tx 1 ~ Tx 11)에 대응하는 트랜잭션 프로포절을 각각 생성한다. 여기서, 트랜잭션 프로포절은 요청된 트랜잭션들(Tx 1 ~ Tx 11)에 대응하는 송금자와 수금자, 송금되는 방향 및 금액 등을 포함하는 트랜잭션의 데이터 형식으로서, 도 2에서 A -> C, A -> D, A -> E, B -> F, B -> G, B -> H, A -> B, C -> D, E -> F, D -> H, G -> H로 이미지화하여 표시하였다.
- [0082] 트랜잭션 패커(110)는 생성된 트랜잭션 프로포절에 트랜잭션 요청에 대응하는 키(key)정보를 추가한다. 여기서, 키 정보는 Ledger Key를 의미하는 것으로, 예를 들어, 도 2의 트랜잭션들(Tx 1 ~ Tx 11)을 구성하는 A, B, C, D, E, F, G, H를 의미한다.
- [0083] 예를 들어, 트랜잭션 패커(110)는 생성된 트랜잭션 프로포절 A -> C에 대응하는 키정보 A 및 C를 추가한 eTP(encapsulated Tx Proposal)를 생성하고, 생성된 eTP를 트랜잭션 에그리게이터(120)에 전달한다. 또한, 트랜잭션 패커(110)는 생성된 트랜잭션 프로포절 A -> D에 대응하는 키정보 A 및 D를 추가한 eTP(encapsulated Tx Proposal)를 생성하고, 생성된 eTP를 트랜잭션 에그리게이터(120)에 전달한다. 이와 같이, 트랜잭션 패커(110)는 일정 시간동안 요청된 트랜잭션에 대해, 순차적으로 트랜잭션들(Tx 1 ~ Tx 11)에 대한 각각의 eTP들을 생성하여 트랜잭션 에그리게이터(120)에 전달한다.
- [0085] 트랜잭션 에그리게이터(120)는 트랜잭션 패커(110)에서 전달받은 상기 트랜잭션 프로포절(eTP)에 포함된 상기 키 정보에 따라 상기 트랜잭션 프로포절에 대응하는 공통 키셋 정보 및 공통 트랜잭션 배치(Transaction batch) 정보를 생성하고, 상기 생성된 트랜잭션 배치정보 별로 구분하여 미리 지정된 실행노드 그룹으로 전송(OUT)한다.
- [0086] 공통 키셋 정보는 복수의 트랜잭션 프로포절들에 포함되는 키 정보들 중 동일한 키를 갖는 것들에 대한 키 집합을 나타낸다. 여기서, 공통 키셋 정보는 전술한 용어 정리에서 R/W Key Set(RWKS)정보를 의미한다.
- [0087] 트랜잭션 배치정보는, 상기 공통 키셋 정보에 포함된 키 정보들에 각각 대응하는 트랜잭션 프로포절들의 집합을 나타낸다. 여기서, 트랜잭션 배치정보는 전술한 용어 정리에서 eTPB를 의미하며, eTPB내의 순서는 eTPID(TxID)에 의해 정의된다.
- [0088] 도 3은 트랜잭션 에그리게이터(120)에서 수행되는 공통 키셋 정보 및 공통 트랜잭션 배치(Transaction batch) 정보를 생성하는 과정을 설명하기 위한 일 예의 참조도이다.
- [0089] 도 3의 (a)는 트랜잭션 패커(110)에서 전달받은 트랜잭션 프로포절(eTP): A -> C를 나타낸다. 도 3의 (a)에는 표시되지 않았지만, 트랜잭션 프로포절(eTP)인 A -> C에는 키 정보에 해당하는 A 및 C의 정보가 포함되어 있다.
- [0090] 도 3의 (b)는 트랜잭션 프로포절(eTP): A -> C에 포함된 키 정보 A 및 C를 이용해 트랜잭션 프로포절(eTP)에 대한 제1 공통 키셋정보인 A 및 C를 생성하였음을 나타내는 참조도이다. 또한, 도 3의 (c)는 공통 키셋 정보인 A 및 C에 대응하는 제1 공통 트랜잭션 배치정보인 A -> C를 예시하는 참조도이다. 도 3의 (b) 및 (c)를 참조하면, 트랜잭션 프로포절(eTP): A -> C에 대응하는 제1 공통 키셋 정보(A, C) 및 제1 공통 트랜잭션 배치정보(A -> C)가 각각 트랜잭션 에그리게이터(120)에 의해 생성됨을 확인할 수 있다.
- [0091] 도 4는 트랜잭션 에그리게이터(120)에서 수행되는 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 다른 예의 참조도이다.

- [0092] 도 4의 (a)는 트랜잭션 패커(110)에서 전달받은 트랜잭션 프로포절(eTP): A → D를 나타낸다. 도 4의 (a)에는 표시되지 않았지만, 트랜잭션 프로포절(eTP)인 A → D에는 키 정보에 해당하는 A 및 D의 정보가 포함되어 있다.
- [0093] 도 4의 (b)는 트랜잭션 프로포절(eTP): A → D에 포함된 키 정보 A 및 D를 이용해 트랜잭션 프로포절(eTP)에 대한 제1 공통 키셋정보인 A, C 및 D를 생성하였음을 나타내는 참조도이다. 제1 공통 키셋 정보는 트랜잭션 프로포절(eTP): A → C와 A → D에 공통인 키 정보 집합에 해당하는 A, C 및 D를 나타낸다. 또한, 도 4의 (c)는 제1 공통 키셋 정보인 A, C 및 D에 대응하는 제1 공통 트랜잭션 배치정보인 A → C, A → D를 예시하는 참조도이다. 도 4의 (b) 및 (c)를 참조하면, 트랜잭션 프로포절(eTP): A → C, A → D 각각에 대응하는 제1 공통 키셋 정보(A, C, D) 및 제1 공통 트랜잭션 배치정보(A → C, A → D)가 각각 트랜잭션 에그리케이터(120)에 의해 생성됨을 확인할 수 있다.
- [0094] 도 5는 트랜잭션 에그리케이터(120)에서 수행되는 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 또다른 예의 참조도이다.
- [0095] 도 5의 (a)는 트랜잭션 패커(110)에서 전달받은 트랜잭션 프로포절(eTP): A → E를 나타낸다. 도 5의 (a)에는 표시되지 않았지만, 트랜잭션 프로포절(eTP)인 A → E에는 키 정보에 해당하는 A 및 E의 정보가 포함되어 있다.
- [0096] 도 5의 (b)는 트랜잭션 프로포절(eTP): A → E에 포함된 키 정보 A 및 E를 이용해 트랜잭션 프로포절(eTP)에 대한 제1 공통 키셋정보인 A, C, D 및 E를 생성하였음을 나타내는 참조도이다. 제1 공통 키셋 정보는 트랜잭션 프로포절(eTP): A → C, A → D 및 A → E에 공통인 키 정보 집합에 해당하는 A, C, D 및 E를 나타낸다. 또한, 도 5의 (c)는 제1 공통 키셋 정보인 A, C, D 및 E에 대응하는 제1 공통 트랜잭션 배치정보인 A → C, A → D, A → E를 예시하는 참조도이다. 도 5의 (b) 및 (c)를 참조하면, 트랜잭션 프로포절(eTP): A → C, A → D 및 A → E 각각에 대응하는 제1 공통 키셋 정보(A, C, D, E) 및 제1 공통 트랜잭션 배치정보(A → C, A → D, 및 A → E)가 각각 트랜잭션 에그리케이터(120)에 의해 생성됨을 확인할 수 있다.
- [0097] 도 6은 트랜잭션 에그리케이터(120)에서 수행되는 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 또다른 예의 참조도이다.
- [0098] 도 6의 (a)는 트랜잭션 패커(110)에서 전달받은 트랜잭션 프로포절(eTP): B → F를 나타낸다. 도 6의 (a)에는 표시되지 않았지만, 트랜잭션 프로포절(eTP)인 B → F에는 키 정보에 해당하는 B 및 F의 정보가 포함되어 있다.
- [0099] 도 6의 (b) 및 도 6의 (c)는 각각 도 5의 (b) 및 도 5의 (c)에 대응하는 것으로, 트랜잭션 패커(110)에서 전달받은 트랜잭션 프로포절(eTP): B → F의 키 정보가 B 및 F에 해당하므로, 제1 공통 키셋 정보와 제1 공통 트랜잭션 배치정보는 변동이 없다.
- [0100] 이에 비해, 도 6의 (d)는 트랜잭션 프로포절(eTP): B → F에 포함된 키 정보 B 및 F를 이용해 트랜잭션 프로포절(eTP)에 대한 제2 공통 키셋정보인 B 및 F를 새로이 생성하였음을 나타내는 참조도이다.
- [0101] 또한, 도 6의 (e)는 제2 공통 키셋 정보인 B 및 F에 대응하는 제2 공통 트랜잭션 배치정보인 B → F를 예시하는 참조도이다. 도 6의 (d) 및 (e)를 참조하면, 트랜잭션 프로포절(eTP): B → F에 대응하는 제2 공통 키셋 정보(B, F) 및 제2 공통 트랜잭션 배치정보(B → F)가 각각 트랜잭션 에그리케이터(120)에 의해 생성됨을 확인할 수 있다.
- [0102] 도 7은 트랜잭션 에그리케이터(120)에서 수행되는 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 또다른 예의 참조도이다.
- [0103] 도 7의 (a)는 트랜잭션 패커(110)에서 전달받은 트랜잭션 프로포절(eTP): B → G를 나타낸다. 도 7의 (a)에는 표시되지 않았지만, 트랜잭션 프로포절(eTP)인 B → G에는 키 정보에 해당하는 B 및 G의 정보가 포함되어 있다.
- [0104] 도 7의 (b) 및 도 7의 (c)는 각각 도 6의 (b) 및 도 6의 (c)에 대응하는 것으로, 트랜잭션 패커(110)에서 전달받은 트랜잭션 프로포절(eTP): B → G의 키 정보가 B 및 G에 해당하므로, 제1 공통 키셋 정보와 제1 공통 트랜잭션 배치정보는 변동이 없다.
- [0105] 이에 비해, 도 7의 (d)는 트랜잭션 프로포절(eTP): B → G에 포함된 키 정보 B 및 G를 이용해 트랜잭션 프로포절(eTP)에 대한 제2 공통 키셋정보인 B, F 및 G를 생성하였음을 나타내는 참조도이다. 제2 공통 키셋 정보는 트랜잭션 프로포절(eTP): B → F 및 B → G에 공통인 키 정보 집합에 해당하는 B, F, G를 나타낸다. 또한, 도 7의 (e)는 제2 공통 키셋 정보인 B, F, G에 대응하는 제2 공통 트랜잭션 배치정보인 B → F 및 B → G를 예시하는 참조도이다. 도 7의 (d) 및 (e)를 참조하면, 트랜잭션 프로포절(eTP): B → F 및 B → G 각각에 대응하는 제

2 공통 키셋 정보(B, F, G) 및 제2 공통 트랜잭션 배치정보(B → F 및 B → G)가 각각 트랜잭션 에그리케이더(120)에 의해 생성됨을 확인할 수 있다.

- [0106] 도 8은 트랜잭션 에그리케이더(120)에서 수행되는 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 또다른 예의 참조도이다.
- [0107] 도 8의 (a)는 트랜잭션 패커(110)에서 전달받은 트랜잭션 프로포절(eTP): B → H를 나타낸다. 도 8의 (a)에는 표시되지 않았지만, 트랜잭션 프로포절(eTP)인 B → H에는 키 정보에 해당하는 B 및 H의 정보가 포함되어 있다.
- [0108] 도 8의 (b) 및 도 8의 (c)는 각각 도 7의 (b) 및 도 7의 (c)에 대응하는 것으로, 트랜잭션 패커(110)에서 전달받은 트랜잭션 프로포절(eTP): B → H의 키 정보가 B 및 H에 해당하므로, 제1 공통 키셋 정보와 제1 공통 트랜잭션 배치정보는 변동이 없다.
- [0109] 이에 비해, 도 8의 (d)는 트랜잭션 프로포절(eTP): B → H에 포함된 키 정보 B 및 H를 이용해 트랜잭션 프로포절(eTP)에 대한 제2 공통 키셋정보인 B, F, G 및 H를 생성하였음을 나타내는 참조도이다. 제2 공통 키셋 정보는 트랜잭션 프로포절(eTP): B → F, B → G, B → H에 공통인 키 정보 집합에 해당하는 B, F, G, H를 나타낸다. 또한, 도 8의 (e)는 제2 공통 키셋 정보인 B, F, G, H에 대응하는 제2 공통 트랜잭션 배치정보인 B → F, B → G, B → H를 예시하는 참조도이다. 도 8의 (d) 및 (e)를 참조하면, 트랜잭션 프로포절(eTP): B → F, B → G, B → H 각각에 대응하는 제2 공통 키셋 정보(B, F, G, H) 및 제2 공통 트랜잭션 배치정보(B → F, B → G, B → H)가 각각 트랜잭션 에그리케이더(120)에 의해 생성됨을 확인할 수 있다.
- [0110] 도 9는 트랜잭션 에그리케이더(120)에서 수행되는 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 또다른 예의 참조도이다.
- [0111] 도 9의 (a)는 트랜잭션 패커(110)에서 전달받은 트랜잭션 프로포절(eTP): A → B를 나타낸다. 도 9의 (a)에는 표시되지 않았지만, 트랜잭션 프로포절(eTP)인 A → B에는 키 정보에 해당하는 A 및 B의 정보가 포함되어 있다.
- [0112] 트랜잭션 프로포절(eTP): A → B에 포함된 키 정보인 A 및 B는 각각 제1 공통 키셋 정보(A, C, D, E)와 제2 공통 키셋 정보(B, F, G, H)에 각각 포함되어 있다. 따라서, 트랜잭션 프로포절(eTP): A → B에 대해서는 제1 공통 키셋 정보(A, C, D, E)와 이에 대응하는 제1 공통 트랜잭션 배치정보(A → C, A → D, 및 A → E)에 포함되도록 분류할 수 없으며, 또한, 제2 공통 키셋 정보(B, F, G, H)와 이에 대응하는 제2 공통 트랜잭션 배치정보(B → F, B → G, B → H)에 포함되도록 분류할 수도 없다.
- [0113] 이에 따라, 트랜잭션 에그리케이더(120)는 키 정보가 복수의 공통 키셋정보들과 교집합을 이루는 경우에는, 해당 키 정보를 포함하는 트랜잭션 프로포절(eTP)에 대해서는 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하지 않고, 다음 주기의 블록체인 실행 시점에 처리할 수 있도록 별도의 임시 저장공간에 저장해 놓는다.
- [0114] 결국, 도 9의 (b), 도 9의 (c), 도 9의 (d) 및 도 9의 (e)는 각각 도 8의 (b), 도 8의 (c), 도 8의 (d) 및 도 8의 (e)에 대응하는 것으로, 제1 및 제2 공통 키셋 정보와 제1 및 제2 공통 트랜잭션 배치정보는 변동이 없다. 또한, 도 9의 (f)는 트랜잭션 프로포절(eTP): A → B에 대해 다음번의 블록체인 처리시점에 실행하기 위해, 별도의 저장공간에 임시 저장한 상태를 나타낸다.
- [0115] 도 10은 트랜잭션 에그리케이더(120)에서 수행되는 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 또다른 예의 참조도이다.
- [0116] 도 10의 (a)는 트랜잭션 패커(110)에서 전달받은 트랜잭션 프로포절(eTP): C → D를 나타낸다. 도 10의 (a)에는 표시되지 않았지만, 트랜잭션 프로포절(eTP)인 C → D에는 키 정보에 해당하는 C 및 D의 정보가 포함되어 있다.
- [0117] 도 10의 (b)는 도 9의 (b)에 대응하는 것으로, 트랜잭션 패커(110)에서 전달받은 트랜잭션 프로포절(eTP): C → D의 키 정보가 C 및 D에 해당하므로, 제1 공통 키셋 정보는 변동이 없다. 다만, 도 10의 (c)는 제1 공통 트랜잭션 배치정보를 나타내는 것으로, 트랜잭션 프로포절(eTP)인 C → D가 추가된 공통 트랜잭션 배치정보로서, A → C, A → D, A → E 및 C → D를 포함하고 있다.
- [0118] 또한, 도 10의 (d) 및 도 10의 (e)는 각각 도 9의 (d) 및 도 9의 (e)에 대응하는 것으로, 트랜잭션 패커(110)에서 전달받은 트랜잭션 프로포절(eTP): C → D의 키 정보가 C 및 D에 해당하므로, 제2 공통 키셋 정보와 제2 공통 트랜잭션 배치정보는 변동이 없다. 또한, 도 10의 (f)는 다음번에 실행되기 위해 임시 저장된 트랜잭션 프로포절(eTP): A → B에 해당한다는 점에서, 변동이 없다.
- [0119] 도 11은 트랜잭션 에그리케이더(120)에서 수행되는 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정

을 설명하기 위한 또다른 예의 참조도이다.

- [0120] 도 11의 (a)는 트랜잭션 패커(110)에서 전달받은 트랜잭션 프로포절(eTP): E → F를 나타낸다. 도 11의 (a)에는 표시되지 않았지만, 트랜잭션 프로포절(eTP)인 E → F에는 키 정보에 해당하는 E 및 F의 정보가 포함되어 있다.
- [0121] 트랜잭션 프로포절(eTP): E → F에 포함된 키 정보인 E 및 F는 각각 제1 공통 키셋 정보(A, C, D, E)와 제2 공통 키셋 정보(B, F, G, H)에 각각 포함되어 있다. 따라서, 트랜잭션 프로포절(eTP): E → F에 대해서는 제1 공통 키셋 정보(A, C, D, E)와 이에 대응하는 제1 공통 트랜잭션 배치정보(A → C, A → D, A → E 및 C → D)에 포함되도록 분류할 수 없으며, 또한, 제2 공통 키셋 정보(B, F, G, H)와 이에 대응하는 제2 공통 트랜잭션 배치정보(B → F, B → G, B → H)에 포함되도록 분류할 수도 없다.
- [0122] 이에 따라, 트랜잭션 에그리게이터(120)는 키 정보가 복수의 공통 키셋 정보들과 교집합을 이루는 경우에는, 해당 키 정보를 포함하는 트랜잭션 프로포절(eTP)에 대해서는 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하지 않고, 다음 주기의 블록체인 실행 시점에 처리할 수 있도록 별도의 임시 저장공간에 저장해 놓는다.
- [0123] 결국, 도 11의 (b), 도 11의 (c), 도 11의 (d) 및 도 11의 (e)는 각각 도 10의 (b), 도 10의 (c), 도 10의 (d) 및 도 10의 (e)에 대응하는 것으로, 제1 및 제2 공통 키셋 정보와 제1 및 제2 공통 트랜잭션 배치정보는 변동이 없다.
- [0124] 또한, 도 11의 (f)는 트랜잭션 프로포절(eTP): E → F에 대해 다음번의 블록체인 처리시점에 실행하기 위해, 별도의 저장공간에 임시 저장한 상태를 나타낸다. 이에 따르면, 이전에 저장되어 있는 트랜잭션 프로포절(eTP): A → B에 추가적으로 트랜잭션 프로포절(eTP): E → F가 저장되어 있음을 확인할 수 있다.
- [0125] 도 12는 트랜잭션 에그리게이터(120)에서 수행되는 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 또다른 예의 참조도이다.
- [0126] 도 12의 (a)는 트랜잭션 패커(110)에서 전달받은 트랜잭션 프로포절(eTP): D → H를 나타낸다. 도 12의 (a)에는 표시되지 않았지만, 트랜잭션 프로포절(eTP)인 D → H에는 키 정보에 해당하는 D 및 H의 정보가 포함되어 있다.
- [0127] 트랜잭션 프로포절(eTP): D → H에 포함된 키 정보인 D 및 H는 각각 제1 공통 키셋 정보(A, C, D, E)와 제2 공통 키셋 정보(B, F, G, H)에 각각 포함되어 있다. 따라서, 트랜잭션 프로포절(eTP): D → H에 대해서는 제1 공통 키셋 정보(A, C, D, E)와 이에 대응하는 제1 공통 트랜잭션 배치정보(A → C, A → D, A → E 및 C → D)에 포함되도록 분류할 수 없으며, 또한, 제2 공통 키셋 정보(B, F, G, H)와 이에 대응하는 제2 공통 트랜잭션 배치정보(B → F, B → G, B → H)에 포함되도록 분류할 수도 없다.
- [0128] 이에 따라, 트랜잭션 에그리게이터(120)는 키 정보가 복수의 공통 키셋 정보들과 교집합을 이루는 경우에는, 해당 키 정보를 포함하는 트랜잭션 프로포절(eTP)에 대해서는 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하지 않고, 다음 주기의 블록체인 실행 시점에 처리할 수 있도록 별도의 임시 저장공간에 저장해 놓는다.
- [0129] 결국, 도 12의 (b), 도 12의 (c), 도 12의 (d) 및 도 12의 (e)는 각각 도 11의 (b), 도 11의 (c), 도 11의 (d) 및 도 11의 (e)에 대응하는 것으로, 제1 및 제2 공통 키셋 정보와 제1 및 제2 공통 트랜잭션 배치정보는 변동이 없다.
- [0130] 또한, 도 12의 (f)는 트랜잭션 프로포절(eTP): D → H에 대해 다음번의 블록체인 처리시점에 실행하기 위해, 별도의 저장공간에 임시 저장한 상태를 나타낸다. 이에 따르면, 이전에 저장되어 있는 트랜잭션 프로포절(eTP): A → B, 및 E → F에 추가적으로 트랜잭션 프로포절(eTP): D → H가 저장되어 있음을 확인할 수 있다.
- [0131] 도 13은 트랜잭션 에그리게이터(120)에서 수행되는 공통 키셋 정보 및 공통 트랜잭션 배치정보를 생성하는 과정을 설명하기 위한 또다른 예의 참조도이다.
- [0132] 도 13의 (a)는 트랜잭션 패커(110)에서 전달받은 트랜잭션 프로포절(eTP): G → H를 나타낸다. 도 13의 (a)에는 표시되지 않았지만, 트랜잭션 프로포절(eTP)인 G → H에는 키 정보에 해당하는 G 및 H의 정보가 포함되어 있다.
- [0133] 도 13의 (b) 및 도 13의 (c)는 각각 도 12의 (b) 및 도 12의 (c)에 대응하는 것으로, 트랜잭션 패커(110)에서 전달받은 트랜잭션 프로포절(eTP): G → H의 키 정보가 G 및 H에 해당하므로, 제1 공통 키셋 정보와 제1 공통 트랜잭션 배치정보는 변동이 없다.
- [0134] 또한, 도 13의 (d)는 도 12의 (d)에 대응하는 것으로, 트랜잭션 패커(110)에서 전달받은 트랜잭션 프로포절(eTP): G → H의 키 정보가 G 및 H에 해당하므로, 제2 공통 키셋 정보는 변동이 없다. 다만, 도 13의 (e)는 제2

공통 트랜잭션 배치정보를 나타내는 것으로, 트랜잭션 프로포절(eTP)인 G → H가 추가된 공통 트랜잭션 배치정보로서, A → C, A → D, A → E 및 G → H를 포함하고 있다. 또한, 도 13의 (f)는 다음번에 실행되기 위해 임시 저장된 트랜잭션 프로포절(eTP): A → B, E → F 및 D → H에 해당한다는 점에서, 변동이 없다.

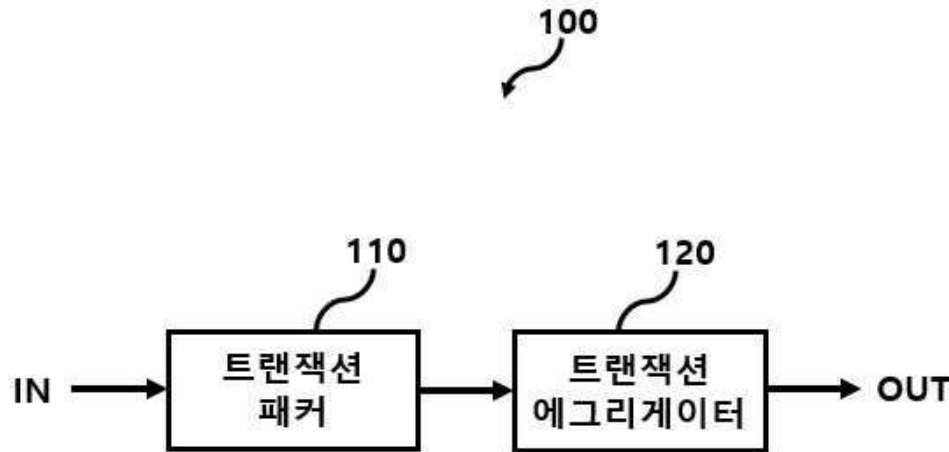
- [0136] 트랜잭션 에그리게이터(120)는 다수의 실행 노드들로부터 상기 실행 노드들 각각의 처리능력정보의 수신에 따라, 상기 공통 트랜잭션 배치정보의 계산을 위한 상기 실행 노드그룹을 지정하고, 상기 지정된 실행 노드그룹으로 상기 공통 트랜잭션 배치정보를 전송한다. 트랜잭션 에그리게이터(120)는 공통 키셋정보 및 공통 트랜잭션 배치정보의 생성 시점에 다수의 실행 노드들로부터 처리능력정보를 수신하여, 이에 따라 실행 노드그룹을 지정할 수 있다.
- [0137] 실행 노드그룹은 복수개의 실행 노드들을 포함하고 있다. 트랜잭션 에그리게이터(120)가 지정된 실행 노드그룹에 속하는 각각의 실행 노드들로 공통 트랜잭션 배치정보를 전송하면, 이를 수신한 실행 노드들은 공통 트랜잭션 배치정보에 저장된 트랜잭션 프로포절에 대한 시뮬레이션(Tx Simulation) 순차적 또는 병렬적으로 실행한다.
- [0138] 실행 노드들 각각은 시뮬레이션 결과에 대해 트랜잭션 에그리게이터(120)로 전송한다. 실행 노드들 각각은 트랜잭션 프로포절에 저장된 정보(트랜잭션 프로포절에 대한 Receptor field의 정보)를 참조하여 Tx Proposal Response를 해당 트랜잭션 에그리게이터(120)로 전송한다.
- [0140] 도 14는 본 발명에 따른 키 정보를 이용한 트랜잭션 분류방법을 설명하기 위한 일 실시예의 플로차트이다.
- [0141] 트랜잭션 패커가 사용자의 트랜잭션 요청에 따른 트랜잭션 정보를 나타내는 트랜잭션 프로포절(Tx Proposal)을 생성하고, 상기 생성된 트랜잭션 프로포절에 상기 트랜잭션 요청에 대응하는 키(key) 정보를 추가한다(S200 단계). 트랜잭션 프로포절은 사용자의 트랜잭션 요청에 대응하는 트랜잭션 정보를 포함하는 것으로, 트랜잭션 에그리게이터(120)에서 사용될 수 있는 트랜잭션의 데이터 형식을 나타낸다.
- [0142] 키(key) 정보는 전술한 용어 정리에서 Ledger Key를 의미한다. 또한, 키(key) 정보는 배열이나 복합키를 포함하는 미확정 형태의 부분 키 정보를 포함할 수 있다. 이는 전술한 용어 정리에서, Partial Ledger Key를 의미한다.
- [0143] 트랜잭션 패커는 사용자의 트랜잭션 요청을 Tx Proposal으로 변환하고, 부가 정보를 추가하여 eTP(encapsulated Tx Proposal)를 생성하고, 생성된 eTP를 트랜잭션 에그리게이터에 전달한다. 여기서, eTP는 기존의 Tx Proposal에 트랜잭션 에그리게이터(120)에 전달하기 위해 부가정보(R/W Key Set Description 정보)가 추가된 Tx Proposal이다.
- [0144] S200 단계 후에, 상기 트랜잭션 패커에서 전달받은 상기 트랜잭션 프로포절에 포함된 상기 키 정보에 따라 상기 트랜잭션 프로포절에 대응하는 공통 키셋 정보 및 공통 트랜잭션 배치(Transaction batch)정보를 생성하고, 상기 생성된 공통 트랜잭션 배치정보 별로 구분하여 미리 지정된 실행 노드그룹으로 전송한다(S202 단계).
- [0145] 공통 키셋 정보는 복수의 트랜잭션 프로포절들에 포함되는 키 정보들 중 동일한 키를 갖는 것들에 대한 키 집합을 나타낸다. 여기서, 공통 키셋 정보는 전술한 용어 정리에서 R/W Key Set(RWKS)정보를 의미한다.
- [0146] 트랜잭션 배치정보는, 상기 공통 키셋 정보에 포함된 키 정보들에 각각 대응하는 트랜잭션 프로포절들의 집합을 나타낸다. 여기서, 트랜잭션 배치정보는 전술한 용어 정리에서 eTPB를 의미하며, eTPB내의 순서는 eTPID(TxID)에 의해 정의된다.
- [0147] 상기 생성된 공통 트랜잭션 배치정보 별로 구분하여 미리 지정된 실행 노드그룹으로 전송하는 단계는, 다수의 실행 노드들로부터 상기 실행 노드들 각각의 처리능력정보의 수신에 따라, 상기 공통 트랜잭션 배치정보의 계산을 위한 상기 실행 노드그룹을 지정하고, 상기 지정된 실행 노드그룹으로 상기 공통 트랜잭션 배치정보를 전송한다.
- [0148] 트랜잭션 에그리게이터는 다수의 실행 노드들로부터 상기 실행 노드들 각각의 처리능력정보의 수신에 따라, 상기 공통 트랜잭션 배치정보의 계산을 위한 상기 실행 노드그룹을 지정하고, 상기 지정된 실행 노드그룹으로 상기 공통 트랜잭션 배치정보를 전송한다. 트랜잭션 에그리게이터는 공통 키셋정보 및 공통 트랜잭션 배치정보의 생성 시점에 다수의 실행 노드들로부터 처리능력정보를 수신하여, 이에 따라 실행 노드그룹을 지정할 수 있다.
- [0149] 실행 노드들 각각은 시뮬레이션 결과에 대해 트랜잭션 에그리게이터로 전송한다. 실행 노드들 각각은 트랜잭션 프로포절에 저장된 정보(트랜잭션 프로포절에 대한 Receptor field의 정보)를 참조하여 Tx Proposal Response를 해당 트랜잭션 에그리게이터로 전송한다.

[0151]

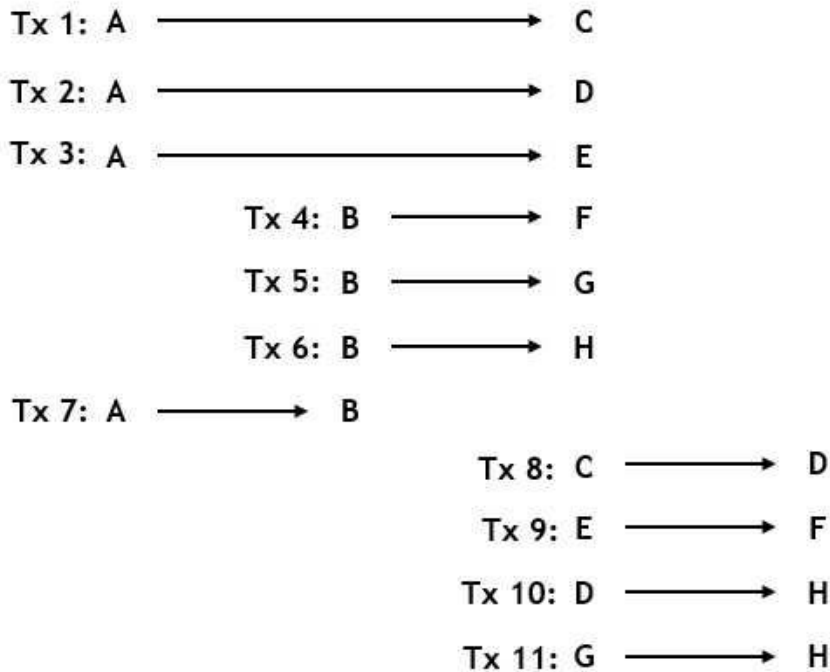
이제까지 본 발명에 대하여 실시예들을 중심으로 살펴보았다. 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자는 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 따라서 본 발명의 범위는 전술한 실시예에 한정되지 않고 특허청구범위에 기재된 내용 및 그와 동등한 범위 내에 있는 다양한 실시 형태가 포함되도록 해석되어야 할 것이다.

도면

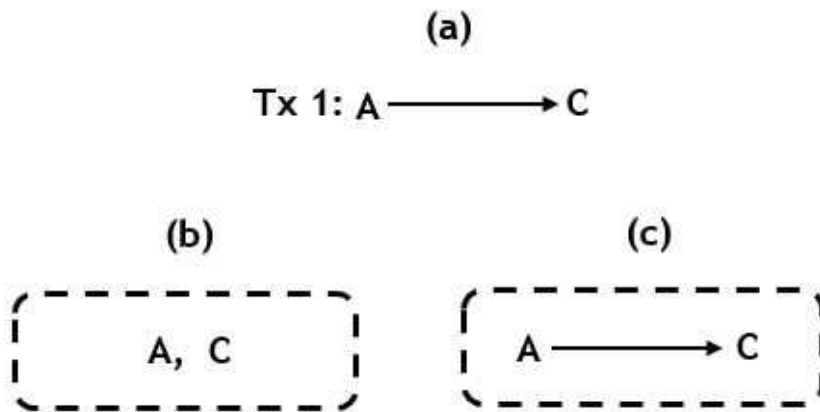
도면1



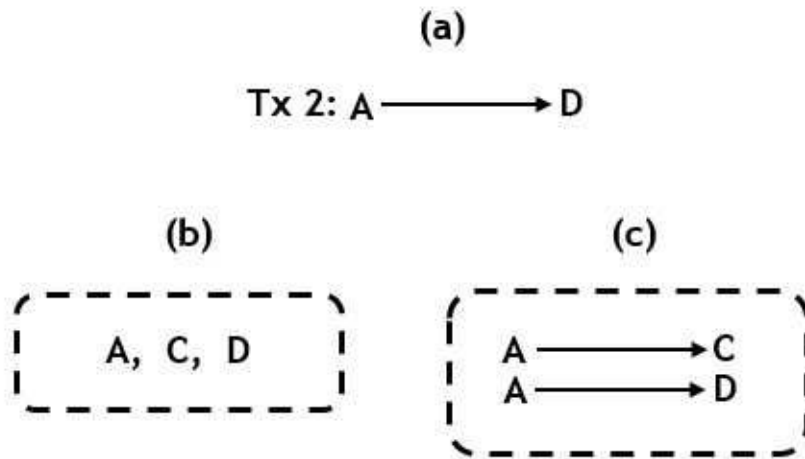
도면2



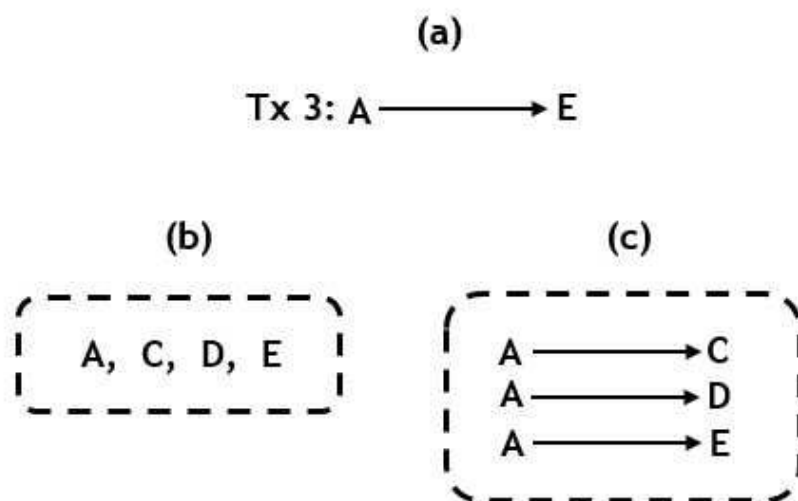
도면3



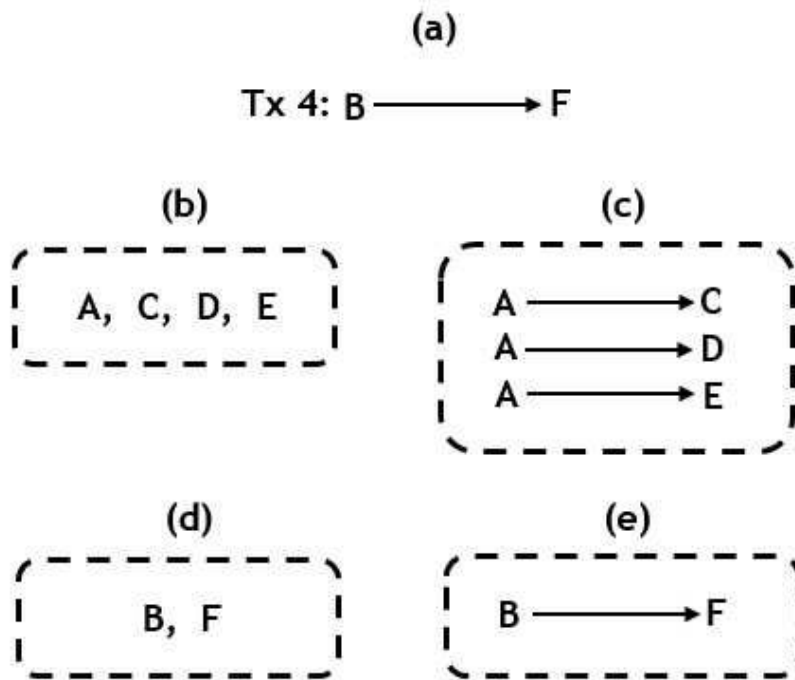
도면4



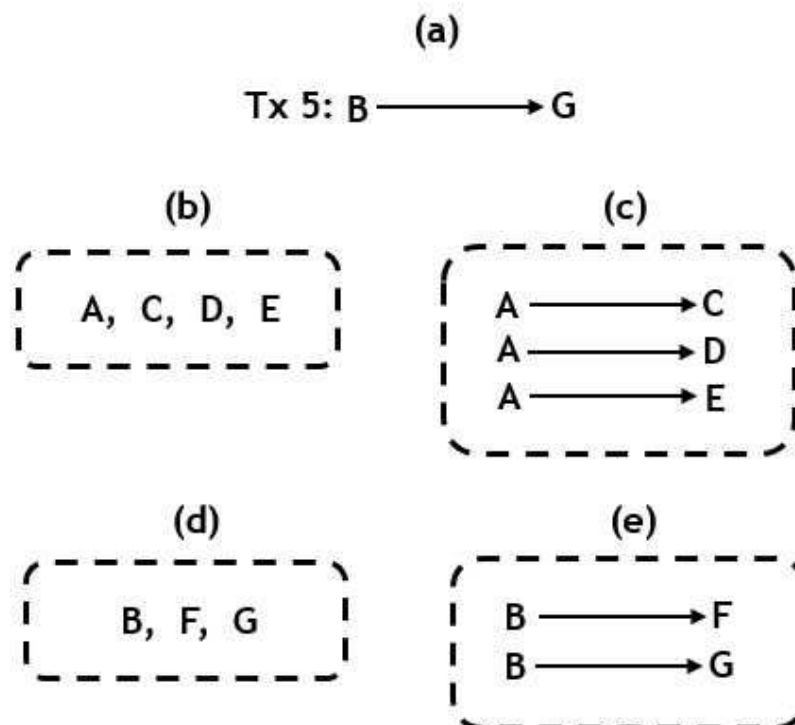
도면5



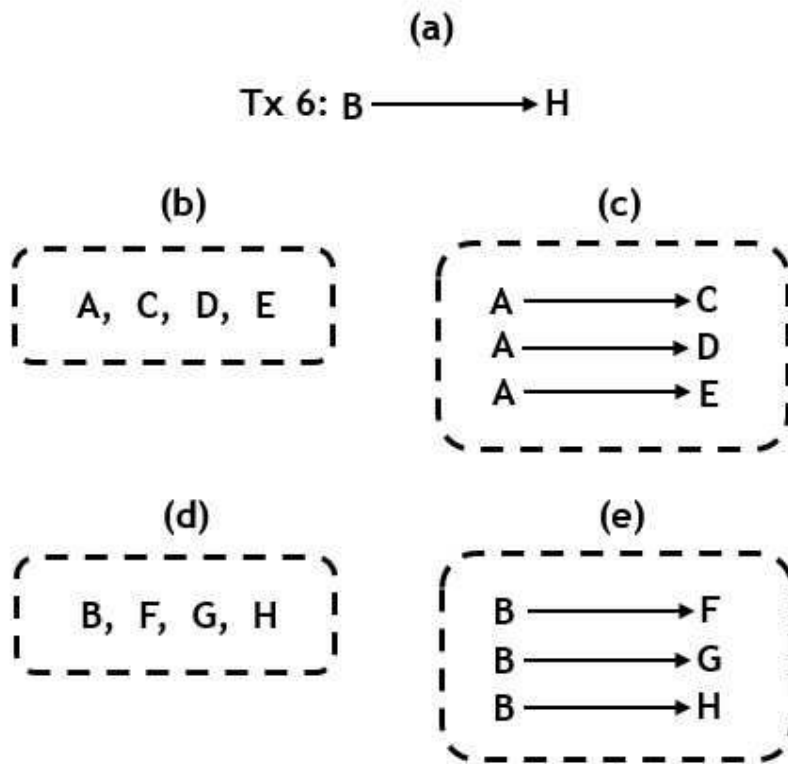
도면6



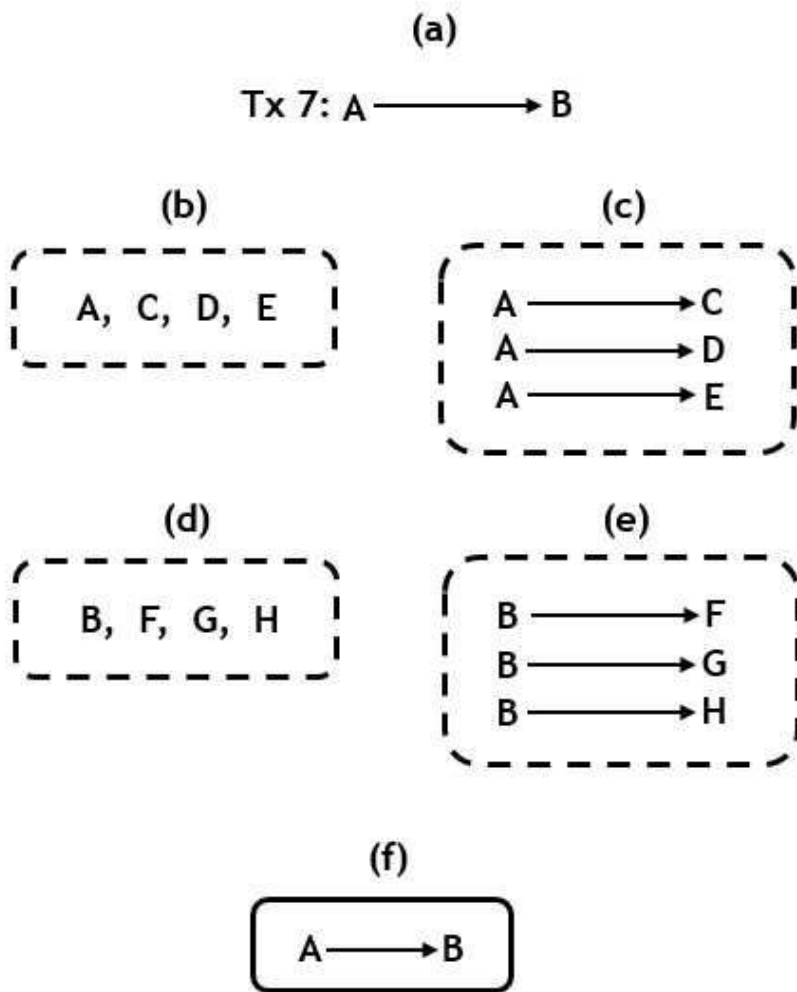
도면7



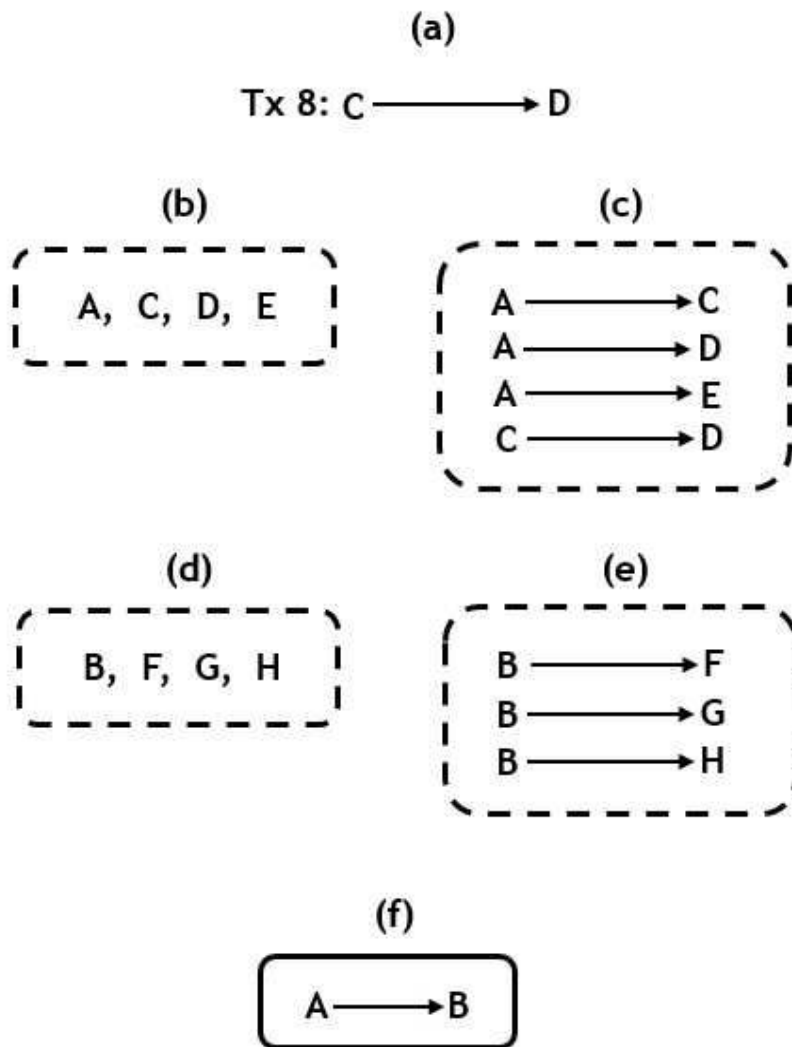
도면8



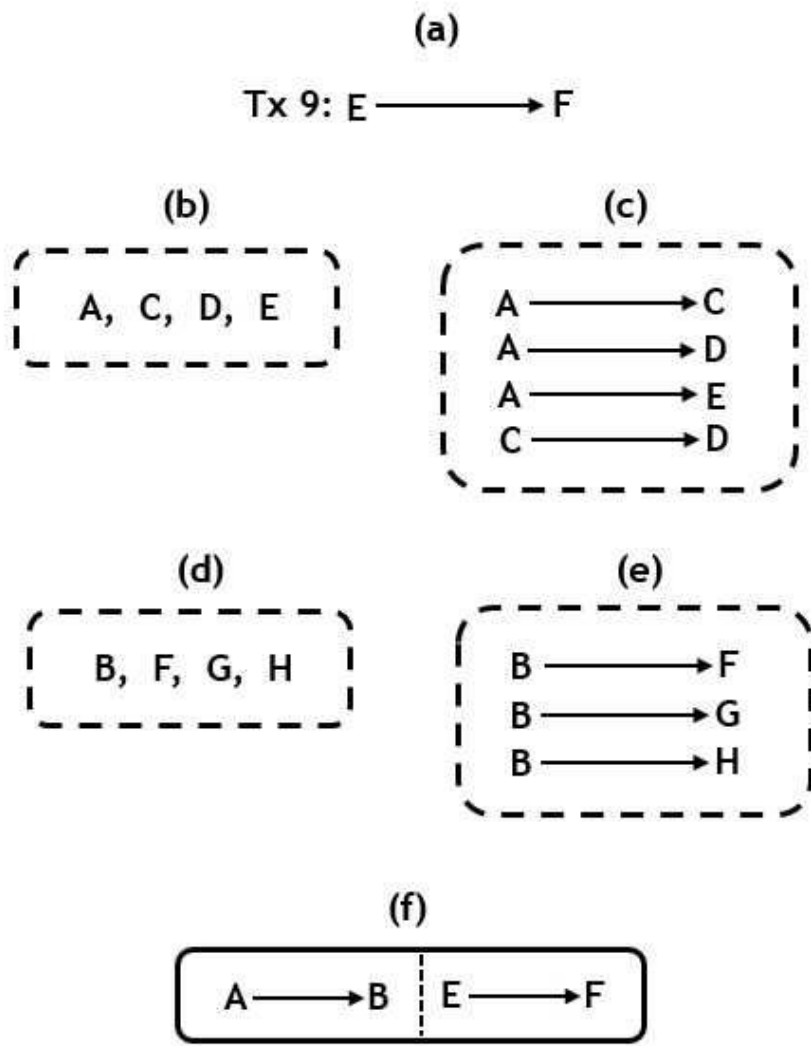
도면9



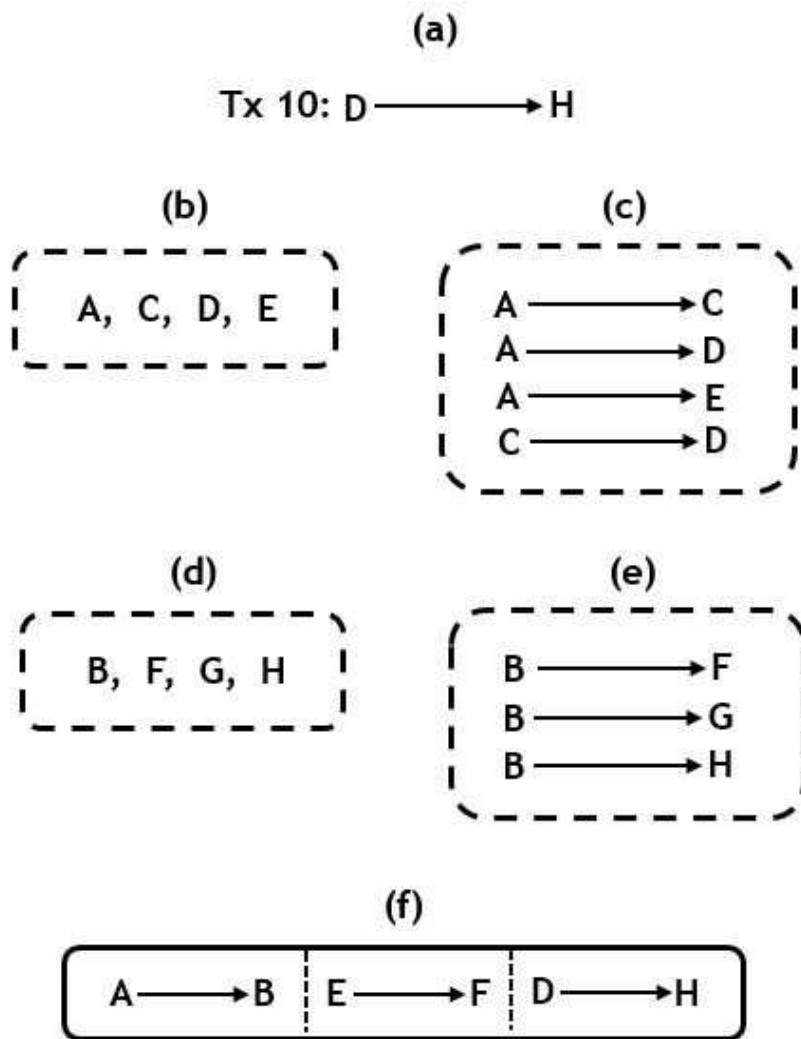
도면10



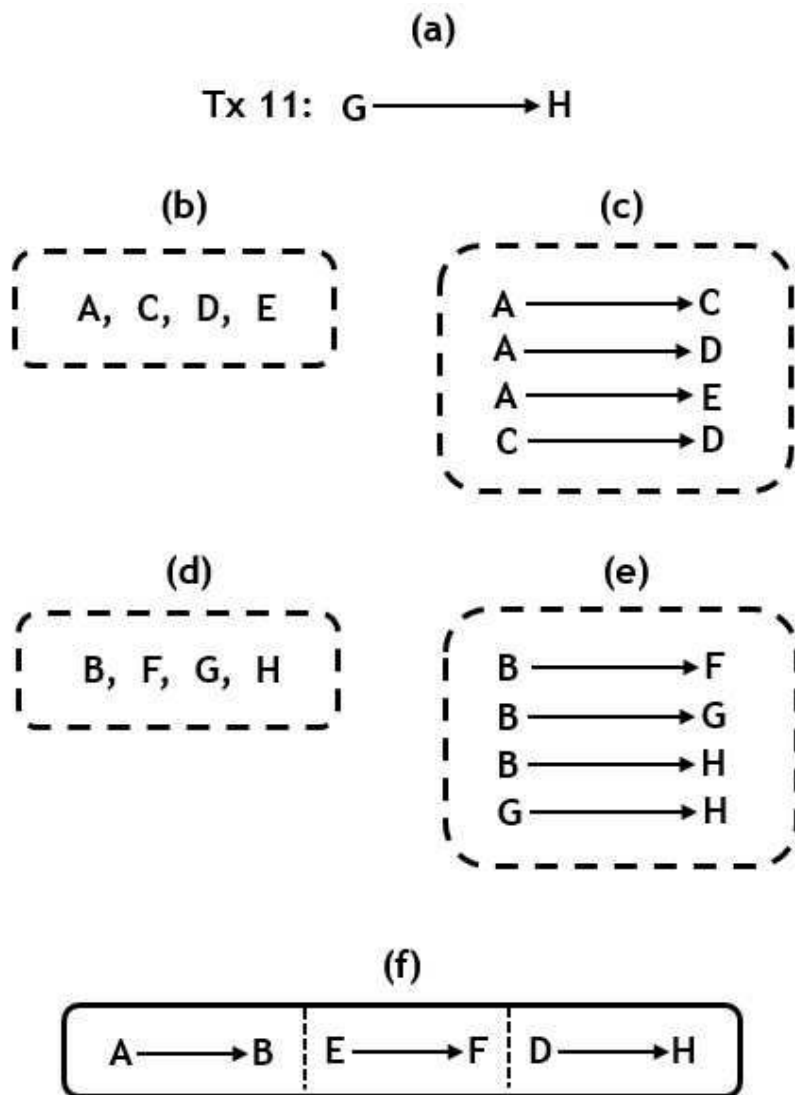
도면11



도면12



도면13



도면14

