공정한 합의 알고리즘 : deb 합의 알고리즘
(A fair consensus algorithm : deb consensus algorithm)

목차

1. 개요

2008년 분산원장(distributed ledger) 개념과 합의 알고리즘인 작업증명(PoW:Proof of Work)을 사용하여 사토시 나까모도가 탈중앙화된(decentralized) P2P 암호화폐시스템인 비트코인(Bitcoin)을 개발하였다. 이후, 2014년 부탈린은 비트코인의 한계를 극복한 "글로벌 신뢰컴퓨터(A trust world computer)"인 이더리움(Ethereum)을 개발하였다.

블록체인(blockchain)의 가장 중요한 핵심 기술은 상호 신뢰하지 않는 노드(Node)들 간의 합의(consensus) 알고리즘이다. 비트코인과 이더리움 모두 합의 알고리즘으로 작업증명 방식을 사용한다. 그러나 작업증명 방식을 사용하는 합의 알고리즘의 경우 노드가 보유한 컴퓨팅 파워(computing power)에 의해 채굴(mining) 확률이 결정된다. 이러한 특성으로 인해 블록체인이 추구하고자 하는 탈중앙화 특성이 약화되는 단점을 가지고 되고, 비트코인의 중앙화 문제가 현실적으로 대두되고 있는 실정이다. 이러한 연유로 이더리움은 현재 합의 알고리즘을 작업증명에서 지분증명(PoS:Proof of Satake) 방식으로 전환하고 있는 실정이다. 그러나 지분증명 방식의 경우에도 노드들의 보유한 지분에 의한 탈중앙화 특성이 지속가능한지에 대한 원천적인 질문을 던지고 있다. 지분증명 방식에 대해 자본주의 문제점을 원천적으로 가지게 될 것이라는 논쟁도 이러한 이유에서 출발한다.

우리는 먼저 블록체인의 핵심 원천기술인 합의 알고리즘에 대한 탈중앙화 특성을 공정성(fairness) 개념으로 정의하여 분석하고자 한다. 간단히 설명하면 합의 알고리즘의 공정성은 채굴을 원하는 노드들의 조건(컴퓨팅 파워, 보유 지분 등)에 따른 채굴 확률의 비례성을 의미한다고 생각할 수 있다.

그리고 공정성을 극대화한 deb 합의 알고리즘을 제안하여 지속가능한 탈중앙화 특성이 유지되는 퍼블릭 블록체인(앤드어스(AndUs) 블록체인)을 제안할 예정이다.

앤드어스 블록체인은 기본적으로 이더리움에 기반한다. 즉, 앤드어스 블록체인은 대표적인 퍼블릭 블록체인(public blockchain)인 이더리움의 구조를 유지하면서, 지속가능한 탈중앙화를 유지하고 속도를 대폭 향상한 퍼블릭 블록체인이다. 현재까지 퍼블릭 블록체인 및 프라이빗 또는 컨소시움 블록체인(private or consortium blockchain) 등 많은 블록체인이 제안되고 있으나, 원래 블록체인의 철학적 특징을 만족하는 것은 이더리움 블록체인이라고 생각하기 때문이다. 특히, 이더리움의 기본 목적인 탈중앙화 P2P 비즈니스 생태계(ecosystem)를 창출하는 인프라로서의 역할이 가장 중요하고 본질적인 블록체인의 철학이기 때문이다.

한편으로는 deb 합의 알고리즘과 기존의 퍼블릭 블록체인에서 사용하는 작업증명 및 지분증명 방식과의 차별성으로 채굴과 암호화폐(cryptocurrency) 발행과의 연관성을 말할 수 있다. 기존의 합의 알고리즘들은 노드들의 채굴 참여를 지속적으로

유지하기 위하여 채굴에 성공한 노드들에게 보상으로 암호화폐 발행 권한을 주는 방식이다.

그러나 deb 합의 알고리즘의 경우, 채굴과 암호화폐 발행과는 연관성이 없다. 즉, 채굴과 암호화폐 발행이 상호 무관한 최초의 퍼블릭 블록체인을 개발하기 위한 합의 알고리즘이다. 채굴자들에게 필요한 보상금을 암호화폐 발행 권한으로 주는 것이 아니고, 채굴에 참여하고자 하는 노드들로 구성될 유료 채굴리그의 참가비의 일부와 및 거래수수료로 보상해주는 방식이다. 이렇게 구성해야 하는 본질적인 이유는 지속가능한 탈중앙화를 위해서 노드들의 채굴 조건과 무관하게 만드는 것과도 연계 된다. 즉, 채굴작업의 공정성을 확보하기 위하여 채굴과정이 모든 노드들에게 공정할 수 있도록 매우 저비용이기 때문에 고액의 보상체계가 필요하지 않다는 것이다.

특히, deb 합의 알고리즘의 경우 기존의 퍼블릭 블록체인과는 달리 포크(fork)가 발생하지 않는 장점 또한 지니고 있다. 이는 블록 생성이 바로 블록의 최종성(finality)을 보장하는 것이다.

2. 합의 알고리즘의 공정성

deb 합의 알고리즘의 목적은 채굴을 원하는 노드들의 조건(컴퓨팅 파워, 보유 지분 등)과 상관 없이 모든 노드들에게 공정한 채굴 확률을 보장함으로서 지속가능한 탈중앙화 특성을 유지하는 것이다.

이를 위해 먼저 합의 알고리즘의 공정성(fairness)을 정의하고 기존 퍼블릭 블록체인들의 공정성을 분석한다.

정의 : 합의 알고리즘의 공정성

> 합의 알고리즘의 공정성이란 노드들의 채굴 확률과 노드들이 가지고 있는 조건(컴퓨팅 파워, 지분 등)들과의 상관 관계로 정의 한다.

예를 들어 비트코인과 이더리움에서 사용하는 작업증명 방식의 경우, 노드들의 채굴 확률은 노드가 보유한 컴퓨팅파워에 위해 결정된다. 즉,

$$\text{노드의 채굴 성공 확률} = \frac{\text{자신이 보유한 컴퓨팅 파워}}{\text{전체 노드들이 보유한 컴퓨팅 파워의 합}}$$

작업증명 방식을 채택하고 있는 비트코인의 경우, 채굴공장 및 그룹의 탄생 등에 따라 일반적인 노드가 채굴에 성공할 확률은 거의 0에 가깝다. 이로 인해 비트코인

은 중앙화되고 있다는 논쟁이 일고 있다.

그리고 이더리움에서 사용하게 될 지분증명 방식의 경우, 노드들의 채굴 확률은 노드가 보유한 지분에 의해 결정된다. 즉,

$$\text{노드의 채굴 성공 확률} = \frac{\text{자신이 보유한 지분}}{\text{암호화폐 총 발행량}}$$

지분증명 방식의 경우는 보유지분에 따른 채굴 확률이 결정됨으로 전형적인 자본의 논리가 적용된다는 문제점이 지적되고 있는 실정이다.

3. deb 합의 알고리즘

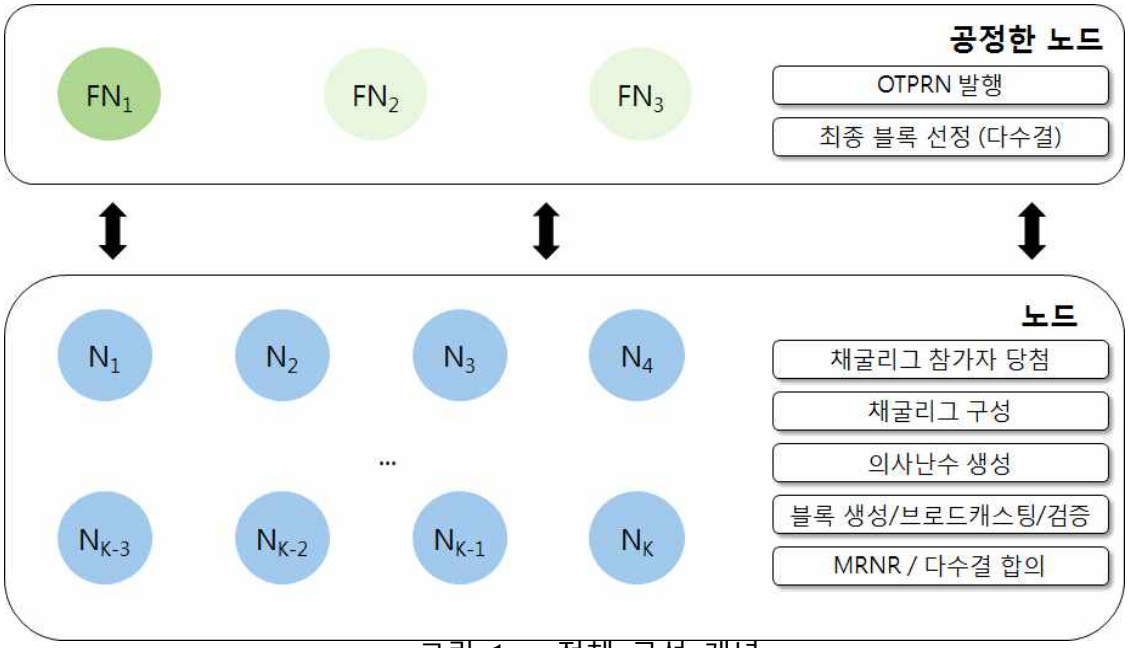
기존의 작업증명 및 지분증명 합의 알고리즘의 경우 채굴 노드가 가지고 있는 컴퓨팅 파워와 보유한 지분에 따라 채굴 노드의 채굴 확률이 비례하는 특성을 가지고 있으며, 이는 채굴 관점에서 블록체인에 참여를 원하는 채굴자들에게 공정(fairness)하지 않다는 것을 말해주고 있다.

deb 합의 알고리즘은 바로 이러한 공정하지 못한 문제점을 해결하여 공정한 채굴 기회를 보장하기 위한 합의 알고리즘이다. 먼저 공정한 채굴 기회를 보장하기 위해서는 채굴을 원하는 모든 노드들에게 주어진 조건(컴퓨팅 파워, 보유 지분 등)에 상관 없이 공정한 채굴 기회를 주어야 한다.

이를 위해 deb 합의 알고리즘은 작업증명과 지분증명 방식과는 달리 공정한 노드(fair node)라는 개념을 도입한다. 물론 P2P 기반의 deb 합의 알고리즘의 특성을 유지하기 위해 공정한 노드의 신뢰성을 가정하지는 않는다. 즉, 공정한 노드는 제3의 신뢰기관(TTP:Trusted Third Party)은 아닌, 단지 P2P 네트워크의 노들들과 협력하여 합의 알고리즘을 지원하는 단순한 특별한 노드라고 생각하면 된다. 공정한 노드의 역할 및 안전성에 대해서는 추후 설명하기로 한다.

deb 합의 알고리즘은 유료 채굴 리그, 최대 난수 규칙(MRNR : Maximum Random Number Rule, 가장 큰 랜덤 넘버) 및 다수결 원칙 등 3가지 기본 원리로 작동된다. 유료 채굴 리그란 채굴을 원하는 노드들 중 특정 수(예, 100명)의 노드들로 구성된 채굴 노드들의 그룹이다. 물론 채굴 리그에 참여를 원하는 노드들은 채굴 리그에 참여하기 위해 현실적으로 충분히 가능한 적은 금액(예, 100원)인 참가비를 지불해야 한다. 그리고 유료 채굴 리그에 참여한 노드들로 구성된 그룹에서 각 노드가 블록을 생성하는 규칙이 최대 난수 규칙이다. 그리고 최종 채굴자를 결정하는 방식, 즉 최종 블록을 결정하는 방식은 공정한 노드와 채굴리그에 참여한 노드들간의 협력을 통한 다수결 원칙으로 이루어진다.

deb 합의 알고리즘의 전체 구성도는 다음과 같다.



< 그림 1 > 전체 구성 개념

공정한 노드의 경우 한 개 또는 다수의 노드로 구성할 수 있다.

3.1 deb 합의 알고리즘 전체 프로세스

deb 합의 알고리즘의 전체 프로세스는 유료 채굴리그 구성, 블록 생성(채굴), 최종 블록 합의 등 크게 3단계로 구성된다.

□ 유료 채굴리그 구성

① 채굴을 원하는 노드는 공정한 노드에게 자신의 접속 정보를 제공한다.

② 공정한 노드는 채굴리그 구성을 위해 모든 노드들에게 $OTPRN$을 배포한다.

③ 채굴리그에 참여를 희망하는 노드는 공정한 노드가 배포한 $OTPRN$을 참조하여 본인이 채굴리그 참여 대상자인지를 판단한다.

④ 채굴리그 참여자로 선정된 채굴노드는 채굴리그 구성을 위해 $OTPRN$을 포함한 $JoinTx$를 생성한다.

⑤ 모든 노드들에게 $JoinTx$를 브로드캐스팅한다.

⑥ 채굴리그 참여자로 선정된 채굴노드들만 $JoinTx$를 참조한다.

□ 블록 생성 (채굴)

① 채굴리그에 참여한 채굴 노드는 최종 블록 선정의 기준이 되는 $difficulty$를 생성한다.

- $difficulty = \{0 \le n \le JoinNonce \mid MAX(CSPRNG(n, OTPRN{\cdot}rand, coinbase, P\_BlockHash))\}$
  ※ 채굴 확률을 균등하게 하기 위해 채굴리그에 신청하였으나, 채굴자로 선정되지 못한 경우 선정되지 못한 경우만큼 복수의 $difficulty$ 생성

② 채굴 노드는 블록 헤더에 $difficulty$를 포함하여 블록을 생성한다.

③ 모든 노드에게 생성된 블록을 브로드캐스팅한다.

□ 합의 알고리즘

블록 합의의 기본 원칙은 가장 큰 수($MRNR: Maximum\ Random\ Number\ Rule$, 가장 큰 랜덤 넘버) 규칙과 노드와 공정한 노드가 협력하여 다수결에 의한 최종 블록 합의 절차이다.

① 노드는 자신이 수신한 블록 중 $difficulty$가 가장 큰 블록을 선택한 뒤 서명하여 공정한 노드에 전송한다.

② 공정한 노드는 다수결 원칙에 따라 전송 받은 블록 중 가장 많은 선택된 블록을 최종 블록으로 결정하여 서명한 후 노드들에게 전송한다.

③ 채굴 노드는 공정한 노드로부터 수신한 블록이 다수에 의해 선택된 블록인지 검증한 뒤 전체 노드들에게 브로드캐스팅한다.

④ 각 노드들은 공정한 노드와 다수가 서명한 블록을 최종 블록으로 인지하고 블록체인에 추가한다.

3.2 유료 채굴 리그 구성 세부 프로세스

안전성 및 효율성을 위해 유료 채굴리그를 구성하는 방법은 공정한 노드와 노드들의 자체적인 인원 조정과 채굴리그 참여 신청으로 진행된다.

□ 유료 채굴리그 참여자 선정

① 채굴을 원하는 노드는 공정한 노드에게 노드 정보를 제공한다.

<표 4-1> enodeCoinbase 구조체

필드명	설명
$enode$	채굴 노드의 enode 값
$coinbase$	채굴에 참여할 계정의 주소
$port$	공정한 노드와 통신할 포트 번호

② 공정한 노드는 블록 생성 주기에 따라 transOTPRN 구조체를 모든 노드에게 배포한다.

<표 4-2> transOTPRN 구조체

필드명	설명
$OTPRN$	채굴 노드들이 채굴 시 참조하는 구조체
Sig	$OTPRN$에 대한 공정한 노드의 서명

<표 4-3> OTPRN 구조체

필드명	설명
num	OTPRN 발행 번호
$rand$	공정한 노드가 주기적으로 배포하는 일회성 의사 난수
$CMiners$	채굴을 수행하기 위해 공정한 노드와 연결을 유지하고 있는 노드의 수
$Timestamp$	공정한 노드의 로컬 시간

③ 채굴 후보 노드들은 공정한 노드가 배포한 $OTPRN$ 구조체를 참조하여 자신이 참가할 수 있는지 파악한다.

(ㄱ) 최대 채굴 참여 인원수를 정의한 시스템 설정 변수 $MMiners$를 제수로 설정

(ㄴ) 채굴 노드는 공정한 노드가 전파한 $OTPRN$ 구조체 중 채굴 의사를 밝힌 전체 채굴 노드 수를 나타내는 $CMiners$를 피제수로 설정

(ㄷ) 두 값을 연산하여 얻은 몫을 Div로 설정

$$Div = CMiners \div MMiners$$

(ㄹ) $enodeCoinbase.coinbase$와 $OPTRN.rand$의 XOR 연산의 합을 랜덤 함수의 시드(SEED)로 사용하여 랜덤 값 도출

$$rand = RAND(\sum_{i=0}^{19}(enodeCoinbase.coinbase[i] \oplus OPTRN.rand[i]))$$

※ $RAND$는 랜덤 함수

(ㅁ) $rand$를 div와 모듈러 연산하여 다음과 같은 조건이 충족될 때 채굴 참여가 가능하다고 판단함

$rand \ \% \ div \ == \ 0$ → 가능

□ 유료 채굴 리그 구성

① 채굴리그에 참가 가능한 채굴노드는 $OTPRN$ 구조체를 포함한 $JoinTx$를 생성하여 모든 노드들에게 브로드캐스팅 한다.

※ $JoinTx$: 노드가 채굴리그에 참여하고자 할 때 발생시키는 채굴리그 참여 신청 트랜잭션

<표 4-4> JoinTx 구조체

필드명	설명
Tx	아래의 필드를 제외하고 이더리움 트랜잭션 구조체와 동일 · $to : Fairnode's\ address$ · $data : JoinTxData$

<표 4-5> JoinTxData 구조체

필드명	설명
$JoinNonce$	계정의 $JoinNonce$에 1을 더한 값
$OtprnHash$	공정한 노드로부터 수신한 $OTPRN$의 해시 값
$FairNodeSig$	$transOTPRN.sig$
$Timestamp$	채굴 노드의 로컬 시간
$NextBlockNum$	채굴할 블록의 번호

② 채굴리그에 참여한 채굴노드들만 $JoinTx$를 수집한다.

③ 채굴노드는 수집한 $JoinTx$를 목록화하여 각자의 채굴리그를 구성한다.

3.3 블록 생성 프로세스 : 채굴 프로세스

공정하고 효율적인 채굴을 위해 공정한 노드와 의사난수($difficulty$)를 활용한다.

□ Difficulty 생성

① 채굴 노드는 참여자 선정 과정에서 공정한 노드로부터 받은 OTPRN 구조체를 참조하여 $difficulty$를 생성한다.

$$difficulty = \{0 \le n \le JoinNonce \mid MAX(CSPRNG(n, OTPRN.rand, coinbase, P\_BlockHash))\}$$

※ $JoinNonce$: $JoinNonce$의 다른 목적으로, 채굴노드가 채굴리그에 참여한 만큼 채굴 확률을 높여주는 기능을 수행함. 이와 같은 목적을 달성하기 위해 $JoinNonce$ 수만큼 다른 $difficulty$를 생성할 수 있고 그 중 가장 큰 값을 블록 생성에 사용할 수 있음.

※ $OTPRN.rand$: 공정한 노드가 배포한 일회성 의사난수로 채굴노드가 $difficulty$를 생성함에 있어 채굴에 유리한 값을 임의로 생성할 수 없도록 함

※ $coinbase$: 채굴노드가 채굴할 때 사용하는 주소로 채굴 노드별로

$difficulty$를 다르게 생성하게 하기 위함

※ $P\_BlockHash$: 이전 블록의 해시 값으로 (i) 공정한 노드가 특정 채굴 노드에게 유리한 $OTPRN.rand$를 배포할 때를 대비하고 (ii) 채굴 노드가 특정 블록에 종속된 하나의 $difficulty$를 생성하도록 하기 위함

② 채굴노드는 자신이 생성한 $difficulty$ 중 가장 큰 $difficulty$를 선택하여 트랜잭션을 생성한다.

$difficulty = MAX(\{difficulty_n\}\ _{0 \le n \le JoinNonce})$

□ 블록 생성 및 브로드캐스팅

① 채굴 노드는 자신의 블록 헤더에 $OTPRN.rand$와 기타 데이터를 참조하여 난수를 생성한 뒤 해당 난수와 자신의 $JoinNonce$를 블록 헤더에 기록한다. 블록 내에 존재하는 $FairNodeSig$와 $Voters$는 향후 공정한 노드에 의해 기록된다.

<표 4-6> 블록 구조체 (아래 필드를 제외하고 이더리움과 동일)

구분	필드명	설명
$Header$	$UncleHash$	제거
	$JoinTxHash$	$JoinTx$ 리스트 해시 값
	$GenTxHash$	Tx 리스트 해시 값
	$JoinReceipt Hash$	$JoinTx$의 receipt 해시 값
	$GenReceipt Hash$	Tx의 receipt 해시 값
	$Difficulty$	채굴 노드가 공정한 노드로부터 수신한 $OTPRN.rand$을 활용하여 생성한 난수 값
	$nonce$	채굴 노드의 $JoinNonce$ 값
	$FairNodeSig$	공정한 노드가 다수의 채굴 노드들이 선택한 블록과 증명 데이터를 포함하여 서명한 값
	$VoterHash$	$Voters$ 해시 값
$Body$	$JoinTxs$	Tx 목록
	$GenTxs$	$JoinTx$ 목록
	$Voters$	해당 블록에 투표한 노드들의 주소와 서명 (복수)

<표 4-7> Voters 구조체

필드명	설명
$addr$	해당 블록에 투표한 채굴 노드의 주소
sig	채굴 노드의 서명
$difficulty$	채굴 노드 자신이 생성한 $difficulty$ 값으로 향후 다수결에 의해 선택된 $difficulty$가 올바른지 검증할 때 사용

② 채굴노드가 수집한 각종 트랜잭션을 블록에 포함한 뒤 블록 생성한다.

③ 채굴노드는 생성된 블록을 다른 채굴 노드들에게 브로드캐스팅한다.

3.4 합의 알고리즘

블록 합의는 기본적으로 $MRNR$과 다수결 원칙에 기반한다. 즉, 최초에 채굴 노드들끼리 각자 생성한 블록을 브로드캐스팅 한 뒤, 자신에게 수신된 블록 중 가장 큰 $Difficulty$가 지정된 블록을 선택($MRNR$)한 뒤 서명하여 공정한 노드에게 전송한다.

공정한 노드는 자신이 수신한 블록 중 다수의 채굴 노드들이 선택한 블록을 선정(다수결 원칙)하여 채굴 노드들의 주소와 서명을 블록 내에 포함시킨 뒤 자체 서명한다. 그리고 공정한 노드가 해당 블록을 채굴 노드들에게 전파하면 채굴 노드들은 해당 블록이 다수결 원칙에 부합하는지, 공정한 노드가 서명했는지 등을 검증한 뒤 원장에 추가한다. 이로써 해당 블록은 최종 블록으로 결정되고, 채굴 노드들은 해당 블록을 네트워크에 전파한다.

□ 유효성 검증 단계

① $OTPRN$ 전파 주기와 블록 생성 주기가 일치하는지 확인한다.

② $OTPRN$ 무결성 및 공정한 노드의 서명을 검증한다.

③ 채굴한 노드가 채굴 리그 참가 가능 대상자인지 확인한다.

④ 채굴 노드가 채굴 리그 참가비를 지불할 수 있는 지 확인한다.

⑤ $Difficulty$가 올바르게 생성되었는지 확인한다.

□ 블록 합의

① 채굴 노드는 수신한 블록 중 $Difficulty$가 가장 큰 블록을 선택($MRNR$)하여 서명한 뒤 공정한 노드에 전송한다.

② 공정한 노드는 전송 받은 블록 중 다수결 원칙에 따라 다수에 의해 선택된 블록을 최종 블록으로 결정하여 서명한 후 노드들에게 전송한다. 향후 검증을 위해 공정한 노드는 해당 블록에 투표한 채굴 노드들의 주소와 서명을 블록에 포함시킨 뒤 서명한다.

③ 공정한 노드는 해당 블록을 채굴 노드들에게 브로드캐스팅한다. 채굴 노드들은 수신한 블록이 다수결 원칙에 부합하는지, 공정한 노드의 서명이 포함되었는지를 검증한 뒤 자신의 원장에 추가하고 해당 블록을 브로드캐스팅한다.

④ 마찬가지로, 위 블록을 수신한 일반 노드들(채굴에 참여하지 않은 노드

들)도 채굴 노드들이 수행한 방식과 동일한 검증 절차를 거친 뒤 해당 블록을 원장에 추가한다.

⑤ 채굴자는 아래와 같이 인센티브를 제공받는다.
인센티브 = 트랜잭션 수수료 + 채굴리그 참가자 전체 참가비

⑥ 채굴리그 참가자들의 채굴 확률을 조정한다.
- 채굴 성공 노드 : $Tx.Join\_Nonce = 0$
- 채굴 실패 노드 : $Tx.Join\_Nonce = Tx.Join\_Nonce + 1$

4. 공정한 노드의 역할 및 신뢰성 문제

비트코인 및 이더리움의 합의 알고리즘은 공정한 노드 개념을 사용하지 않는다. 그러나 deb 합의 알고리즘의 경우 지속가능한 탈중앙화 특성을 유지하기 위해 공정한 노드 개념을 도입하였다. 물론 deb 합의 알고리즘이 동작하기 위해서 공정한 노드의 신뢰성을 가정하지 않는다.

공정한 노드는 유료 채굴리그 구성의 효율성, 블록 합의 및 최종성 협력을 위한 역할만을 담당한다.

□ 공정한 노드의 역할

① 유료 채굴리그 참여자의 랜덤한 선정

② 노드들과의 상호 견제를 통한 최종 블록 합의 협력

가장 중요한 것은 deb 합의 알고리즘은 공정한 노드의 신뢰성에 의존하지 않는다. 공정한 노드와 블록체인 노드들간의 상호 견제를 통해 공정한 노드의 신뢰성 보장 없이도 블록체인의 안전성을 확보할 수 있다.

공정한 노드를 이용하여 deb 합의 알고리즘의 공정성은 다음과 같이 생각할 수 있다.

$$\text{노드의 채굴 확률} = \frac{1}{\text{채굴 희망 전체 노드수}}$$

5. 성능

deb 합의 알고리즘의 성능은 유료 채굴리그 구성 수와 블록생성 주기 등에 따라 동적으로 결정될 수 있다.

예를 들어, 채굴 리그 인원 수 100명인 경우 예상되는 성능은 다음과 같다.

<표 4-8> deb 합의 알고리즘의 성능

	deb 합의 알고리즘
Block Size	4.5MB ~ 9MB
TPS	1,200 TPS
생성주기	5초 ~ 30초

현재 채굴자의 PC환경을 최소화(일반 가정 PC)하였을 경우 200 TPS가 실현되었으며, 채굴자의 네트워크 환경을 개선하면, 블록생성 시간을 5초로 줄이고, 8G 메모리를 사용하여도 200 TPS에서 이더리움 속도의 100배 이상인 1,200 TPS로 개선이 된다.

특히, 블록 생성 시간을 줄이기 위해 공정한 노드와 유료 채굴 리그 노드들 간의 네트워크 접속부하를 줄이면 성능은 더 개선할 수 있다. 일례로 한번 구성된 유료 채굴 리그의 블록 생성 숫자를 10개로 한다면 성능은 10,000 TPS 이상이 된다.

또한 이더리움에서 성능 개선을 위해 적용한 샤딩기술 한가지를 단순 적용만 하더라고 20,000 TPS 이상의 성능을 확보할 수 있으며, 2020년 말에는 최소 100,000 TPS 성능을 실현하고자 한다.

6. deb 합의 알고리즘 특징

deb 합의 알고리즘의 목적은 현재의 합의 알고리즘의 불공정성으로 인해 발생할 수 있는 블록체인 합의 알고리즘의 중앙화 문제를 해결하는 것이다. 즉, 기존의 합의 알고리즘인 작업증명 방식, 지분증명 방식과 deb 합의 알고리즘의 가장 큰 차이점은 지속가능한 탈중앙화를 유지할 수 있다는 것이다. 이는 채굴을 원하는 노드들의 조건들에 의존하지 않는 공정한 합의 알고리즘이라는 것을 의미한다.

또한 기존의 퍼블릭 블록체인의 합의알고리즘의 경우 블록을 생성하는 채굴과 암호화폐 발행이 연계되어 있으나, deb 합의 알고리즘의 경우 채굴과 암호화폐 발행이 무관하다는 것이다. 즉, 초기 발행한 암호화폐 발행량이 바로 총 통화량이 된다는 것을 의미한다.

이는 암호화폐 발행 권한을 독점하면서도 퍼블릭 블록체인을 구성할 수 있게 하는 최초의 합의 알고리즘이다.

한편으로 deb 합의 알고리즘의 장점으로는 포크(fork)가 일어나지 않아 최종성이 1 블록이면 달성되는 장점이 있다.

□ deb 합의 알고리즘 특징

① 지속가능한 탈중앙화 특성 유지 (공정성)

※ 퍼블릭 비허가형 블록체인 중 유일

② 채굴과 암호화폐 발행 무관 (암호화폐 발행 독점 가능)

① 포크 없는 1블록의 최종성 보증

※ 퍼블릭 비허가형 블록체인 중 최고

② 1,200 TPS 이상의 고속 성능

※ 퍼블릭 비허가형 블록체인 중 세계 최고 속도

7 .결론

deb 합의 알고리즘은 기존의 합의알고리즘인 작업증명 및 지분 증명 방식의 특성(노드들의 조건)으로 인한 중앙화 문제를 해결한 최초의 합의 알고리즘이다. 이는 블록체인의 원래 목적인 지속가능한 탈중앙화를 달성할 수 있는 핵심적인 개념이다.

또한, 퍼블릭 블록체인이 작동하기 위한 보상체계로 새로운 암호화폐 발행과 연계하지 않는 최초의 퍼블릭 블록체인이다. 이는 채굴자의 채굴 비용을 최소화함으로서 지속가능한 탈중앙화 특성과 연계되어 있다.

특히, deb 합의 알고리즘은 현재까지 제안된 퍼블릭 블록체인 중 지속가능한 탈중앙화를 유지하면서도 성능이 가장 우수한 퍼블릭 블록체인 앤드어스 블록체인(AndUs blockchain)을 실현할 수 있게 하는 핵심 원천기술이다.

< 표 2 > 주요 퍼블릭 블록체인 성능 비교

	비트코인	이더리움	앤드어스 블록체인
합의 알고리즘	작업증명	작업증명	deb 합의 알고리즘
TPS	7	12~15	1,200 이상
최종성	10분	약 3분	5초 ~ 30초

우리는 공정한 조건에서 동일한 채굴 확률을 갖게 되는 지속가능한 탈중앙화를 유지하게 될 것이다. deb 합의알고리즘을 기반으로 한 "앤드어스체인(Anduschain)"은 진정한 의미에서의 공정성을 갖춘 고속 퍼블릭 블록체인이다.

변경 내역

2019.05.31. - 문서버전 0.91

· 변경 원인 : 채굴 참여 트랜잭션은 수수료가 부과되지 않기 때문에 일반 트랜잭션 대비 블록 포함 우선순위가 낮아 블록에 포함되지 않음

· 해결 방안 : 채굴 참여 트랜잭션의 처리 우선순위를 최우선으로 지정

· 변경 위치 : 블록 구조체

2020.04.30. - 문서버전 0.95

· 변경 원인 : 속도 개선(1,200 TPS 이상)

· 해결 방안 : 채굴리그 구성 방안 변경 및 속도 성능 개선 방향 제안(샤딩기술 적용 시 20,000 TPS 이상)

· 변경 위치 : 5. 성능