블록체인에서의 합의 알고리듬

실시간지급결제 환경에서 지분증명(PoS)의 적용 가능성을 중심으로[*]

전 주 용 †, 정 은 진 ‡, 성 준 이 §, 이 동 영 **

1. Introduction

2. 블록체인과 포크

3. 블록체인과 합의

4. 합의의 구현 방향 및 활용 가능성

5. 정리 및 제언

부록 1. 이더리움 2.0 지분증명 알고리듬

부록 2. 투표 패러독스

참고문헌

[*] 본 연구는 한국은행의 지원을 받아 수행되었으며, 본 연구에 기술된 내용은 연구진의 견해로 한국은행의 공식입장과는 무관함

† 동국대학교 경제학과 (jooyong@dongguk.edu)

‡ Department of Computer Science, University of San Francisco (ejung@cs.usfca.edu)

§ 한국은행 전산정보국 과장

** 한국은행 전산정보국 조사역

초록

본 연구는 블록체인에서 투표를 이용한 지분증명(Proof of Stakes, PoS) 방식 합의 알고리듬에 대해 이더리움(Ethereum) 2.0에서 제안된 내용들을 중심으로 공학 및 경제학적 관점에서 살펴보고 실시간 지급결제환경에서 지분증명 방식 블록체인의 적용 가능성에 대하여 살펴본다.

비트코인을 비롯하여 현재 대부분의 블록체인에서 합의 도출을 위해 이용되는 작업증명(Proof of Works, PoW) 방식의 경우 처리 속도 및 운용 비용 측면에서 결제망과 같은 실시간 대용량 처리에 적용하기에는 한계를 보인다. 지분증명 방식에서는 정격(justified) 블록을 선정하기 위하여 지분을 가진 노드들이 지분에 비례한 확률로 거래를 처리하고 신규 블록을 생성할 권한을 갖게 되며, 그 결과는 참가 노드들의 투표를 이용한 합의로 완결된다. 지분증명 방식 블록체인의 경우 투표를 이용하기 때문에 사회 선택(social choice) 이론에서 알려진 의제 조작 (agenda manipulation) 혹은 스포일러(spoiler) 후보를 이용한 선택 결과 조작 가능성에 대해 고려해야 한다.

블록체인 시스템에서는 시빌 공격(Sybil)을 막기 위하여 노드별로 연산 부하 혹은 경제적 지분이 요구되는 구조이며, 기수적(cardinal) 보상구조 덕분에, 의제 조작을 통한 투표결과의 조작 가능성은 높지 않지만 정보전달 지연이나 유실 등 블록체인 네트워크의 불완전성이 정보 캐스케이드(information cascade) 문제가 결합될 경우 지분비중이 높은 노드가 낮은 노드를 대상으로 특정 선택을 유도할 수 있는 가능성이 존재할 수 있다. 이를 막기 위하여 본 연구는 지분율과 투표 타이밍을 동시에 고려하는 스코어링(scoring) 방식의 보상을 적용될 수 있음을 제시한다.

끝으로, 지분증명 금융기관 간 실시간 거액결제망 에서는 이더리움과 같은 공개 지분증명 방식의 블록체인의 이용은 지분과 컴퓨터 자원의 균형을 맞추는 과정에서 공개 블록체인 방식에서 기대하는 다수의 참가자에 의한 분산 원장 구현이라는 목표가 구현되기 어려우며, 폐쇄형(컨소시움) 방식의 경우 위기 상황에서 노드를 운용하는 금융기관의 유동성 확보를 놓고 이해가 충돌할 경우 거래 처리가 지연되는 문제가 발생할 수 있으며, 중앙은행의 개입이 이를 완화할 수 있지만 탈중앙화 특성은 그 과정에서 훼손될 수 밖에 없음을 확인하였다. 결과적으로 현 시점에서 본다면 작업증명 방식 블록체인과 마찬가지로 지분증명 방식 블록체인 역시 결제망에 이용하기는 어렵다고 판단한다.

0. Introduction

블록체인을 바라보는 관점은 여러가지가 있을 수 있지만 하나의 방법은 네트워크로 연결된 시스템 내에서 일어나는 모든 거래(transaction)에 대한 기록을 시스템 내에 있는 복수의 노드에 걸쳐 저장(분산원장, distributed ledger)하고 있는 분산 데이터베이스 시스템(distributed database system)의 일종으로 이해하는 것이다. 분산 컴퓨팅 시스템은 자원 공유(resource sharing), 개방성(openness), 동시성(concurrency), 무결성(integrity), 확장성(scalability), 장애 허용성(fault tolerance), 투명성(transparency) 등의 특성을 제공할 수 있어야 하기 때문에 이전에는 미리 신뢰할 수 있는 노드(node, 개별 컴퓨팅 시스템) 사이에서만 구축되는 것이 당연하게 여겨졌다. 그러나, 블록체인 원리를 최초로 구현한 비트코인, 그리고 송금을 넘어 구현 가능한 거래의 범위를 어플리케이션(스마트 계약) 수준으로 확장시킨 이더리움의 등장은 블록체인 네트워크와 같이 사전적 신뢰가 구축되지 않은 노드들로 이루어진 분산 시스템의 구현 또한 가능할 수 있음을 보여주었다.

거래 처리(transaction processing) 관점에서 본다면 분산화되어 처리되고 분산원장에 기록된 거래들은 직렬화(serialization)[1] 요건을 만족해야 한다. 즉, 단일한 데이터베이스 시스템에서 순차적으로 거래가 처리된 결과와 동등해야 한다. 블록체인과 같이 사전적 신뢰가 구축되지 않은 노드로 이루어진 분산 시스템에서 거래 처리 분산화(decentralization)에 필수적인 요소는 거래 내용의 일방 암호화를 통하여 생성되는 (i) 해시 체인(hash chain)과 더불어 유효한 해시 체인에 대한 참가자들의 (ii) 합의(consensus)를 통한 항상성(consistency)의 유지이다.

블록체인을 이용한 거래 처리에서 가장 문제가 되는 것은 단일한 (정격) 블록에서 두 개 이상의 신규 블록이 갈라져서 생성되는 분화(fork) 상황이다. 이는 이중 지불 등과 같은 특정 노드의 악의적인 의도가 없다고 하더라도 네트워크의 불완전성으로 인해 "동시"로 간주될 수 있는 시간에 복수의 블록에 생성되거나, 혹은 블록 생성 여부가 어떤 이유에서든 전체 네트워크에 전파되지 못한 상황에서 또 다른 신규 블록을 만들어낸 경우 등이 발생 원인으로 작용할 수 있다. 정격 체인에 포함된 거래 내용은 직렬화 요건을 만족하여야 하기 때문에 포크가 일어날 경우 하나의 블록만 정격 블록으로 인정해야 한다.

현재까지 거래 유효성을 검증하고 합의하기 위하여 가장 널리 쓰여온 작업증명(Proof-of-Work) 방식에서는 포크가 발생할 경우 어떠한 블록을 선택할 것인지를 다음 번 채굴에 성공한 노드의 선택에 따라 결정한다.[2] 반면, 지분증명(Proof-of-Stake, PoS) 방식의 경우 합의에 도달하려면 (i) 누가 다음 블록에 저장될 내용들을 결정하며 (ii) 그 내용들에 대해 동의를 얻어낼 것인지가 해결되어야 하며, 이를 위해 투표 방식을 채택한다. 투표를 통하여 합의를 도출하고자 할 경우 (i) 리더를 선출하고 신규 블록에 어떠한 내용을 저장할지 정하거나 (leader election) (ii) 이용자들이 신규 블록 후보들을 제안하고 투표를 통해 블록을 결정(value election)할 수 있다.

---

[1] Gray, J. (1981, September). The transaction concept: Virtues and limitations. In *VLDB* (Vol. 81, pp. 144-154).

[2] 다만, 완전히 임의적인 것은 아니며 longest-chain (비트코인) 혹은 heaviest-chain (이더리움) 으로 알려진 규칙을 따르도록 설계되어 있다.

지분증명 방식의 경우 작업증명 방식이 갖는 에너지 비효율성 및 완결성(finality) 미제공 등의 문제를 해결하기 위하여 제시되었지만, 이를 구현하기 위해서 해결해야 할 문제들은 작업증명 방식에 비해 많고 제시된 해결책들도 아직 실제 상황에서 충분한 검증이 이루어지지 않았다. 작업증명방식의 경우 (i) 거래 처리 완료 및 블록 추가에 따른 보상을 노리며 서로 경쟁하는 채굴자들의 기대 이익이 0이라는(zero-profit) 조건을 만족하고 (ii) 분산원장 시스템을 대상으로 한 "과반수 공격(majority attack)" 시도에 따른 비용이 편익을 상회하여야 한다는 점을 확인하여야 한다 (Budish, 2018). 이에 비해 지분증명방식의 경우 (i) Who gets to vote? (ii) How to find/advertise candidate blocks (iii) How to decide which block to vote on (iv) How to cast a vote with integrity (no one tampers my vote) (v) How to tally the votes (no one can vote twice with one stake) (vi) When and who to close the voting process 등 보다 많은 문제를 처리해야 한다 (Jung, 2018). 무엇보다도, Arrow의 불가능성 정리, Gibbard and Satterthwaite 정리 등은 3개 이상의 후보 중에서 선택하는 "일반적인" 사회적 선택 상황에서 경제적으로 효율적이고, 합리적이고, 평등하면서도, 조작이 불가능하고, 독재적이지 않은 투표 제도를 설계하는 것은 불가능하다는 것을 보인다.[3] 실제로 이더리움의 채굴 기록을 보더라도 하루에 30개의 3분화(3-way fork)가 발생하는 경우도 있으며, 심지어는 5분화(5-way fork)가 일어나는 경우도 관찰된다. 따라서, 투표 알고리듬을 설계할 때에는 이러한 문제에 대해 고려해야 함을 알 수 있다.

또한, 지분증명 방식은 흔히 Nothing-at-Stake 문제라고 알려진 문제가 발생할 수 있다. 이는 하나의 노드가 복수의 후보에 대해 투표를 수행하더라도 기댓값이 양이 될 수 있고 그로 인하여 투표를 통하여 정상적인 정보 수집 및 정격 블록 선택이 불가능하게 될 수 있다는 문제이다. 이로 인하여, Casper the Friendly Finality Gadget (Buterin, 2018) 등과 같이 정격으로 선정되지 않은 블록에 투표할 경우 정격 블록에 투표할 때 받게되는 보상금액보다 큰 페널티를 부과하는 방식, EOS 등과 같이 투표가 가능한 노드를 아예 블록체인 시스템 외부에서 정하는 Delegated PoS (DPoS) 방식 등이 제안되었다 (Larimer, 2017). Saleh (2021)는 PoS에서 적절한 수준의 보상 구조(schedule)를 통해 가능한 신속한 합의가 이루어질 뿐만 아니라 영속적인 포크(fork)가 발생할 가능성이 배제되는 균형이 존재할 수 있음을 보인다. Saleh(2021)는 이러한 결과는 PoW와 달리 PoS에서는 검증인(validator)도 이해 관계자가 되기 때문이라고 주장한다. 그러나, 이러한 방식이 실제로 네트워크 상에서 구현될 경우 지연으로 인하여 투표 참가자가 유력 참가자의 결정을 기다렸다가 이를 추종하는 행동이 일어날 수 있다. 특히 투표와 같이 관찰값이 단속(discrete)적일 경우에는 information cascading 및 정보의 손실로 인한 시스템 전반적인 비효율적 결정이 발생할 가능성이 존재한다 (Bikhchandani et al., 1998). 또한, 투표 가능 노드가 소수가 될 경우 노드 간 답합에 보다 취약해질 수 있으며, 대형 노드의 독점화 경향과 그로 인하여 부익부, 빈익빈 현상이 강화될 가능성도 존재한다.

본 연구에서는 투표를 통한 정격 블록 혹은 체인의 선정에 있어서 어떠한 문제들에 대한 고려가 필요한지 알아보고 이에 대한 대응 방안을 제시하고자 한다. 보다 구체적으로는, 현재 가장 널리 이용되는 블록체인인 이더리움(Ethereum)의 정격 블록 특성을 분석하고, GHOST, Casper the FFT 등 현재 제안된 이더리움 지분증명 방식 합의 알고리듬에 대하여 사회선택(social choice) 및 사회학습(social

---

[3] 반면 모든 대안들에 대해 모든 투표자들이 단일 정점(single-peaked) 선호를 갖는 등 특정한 조건이 추가로 주어진다면 위에서 주어진 원칙들을 만족하는 사회적 선택 방식(투표 제도)을 설계할 수 있다.

learning) 이론 측면에서 살펴보고 예상되는 문제점을 지적하며, 그에 대한 대응 방안을 제시하고자 한다. 끝으로, 이더리움 블록체인의 금융기관 간 및 중앙은행 결제망 적용에 대한 타당성에 대하여 의견을 제시하고자 한다.

1. 블록체인과 포크

1.1 포크란 무엇인가

블록체인의 위변조 저항성과 데이터무결성은 해시함수를 이용해서 블록을 일방향으로만 연결하는 데이터구조에서 온다. 블록체인에 쓰이는 해시함수는 특별한 연산함수로, 입력값 (블록의 내용), 출력값 (다음 블록에 포함될 숫자) 이 알려져 있을 때 같은 출력값을 갖는 다른 입력값 (다른 내용의 블록)을 찾기 매우 어렵다. 마치 메신저에서 이전 메시지를 인용해서 답장을 하면 나중 메세지만 보고도 이전 메세지를 알 수 있어서 이전 메시지를 수정하거나 삭제해도 소용이 없는 것과 비슷한 이치다.

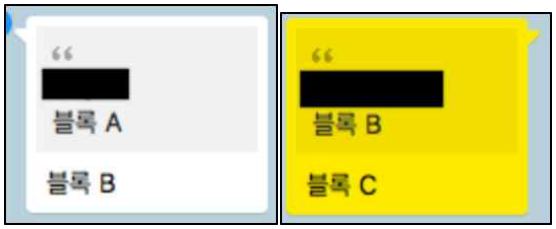


그림 1 신규 블록의 생성

하지만 데이터의 위변조 저항성과 무결성은 나중에 생성된 블록에서 과거에 생성된 블록을 확인할 때에만 사용가능한 것으로, 말하자면 "역사를 고쳐쓸 수는 없다"는 의미이지, "과거를 알고 있으면 미래를 알 수 있다"는 의미는 아니다. 블록 B를 만든 사람은 블록 A의 내용을 확인할 수 있지만, 아래 그림의 블록 D와 위 그림의 블록 C중 어느 쪽이 맞는지는 알 수 없다. 이렇게 한 블록 (블록 B) 에서 두 개의 블록 (블록 C와 블록 D) 이 이어지는 것을 갈림길이라는 의미에서 **포크(fork)**라고 한다.

그림 2 포크(fork)의 발생

1.2 포크가 왜 문제인가

블록체인은 거래내역을 저장하는 분산원장(distributed ledger)으로 가장 많이 이용되고 있다. 암호화폐의 거래내역이든, 거대 슈퍼마켓 체인의 야채 거래내역이든 블록체인에 저장하게 되면 중앙화된 서버 없이도 위변조 없는 원장을 모든 참여자가 공유할 수 있게 된다. 새로운 거래내역은 블록체인에 쓰여지기 전에 블록생성자들의 검증 과정을 거친다. 예를 들어 지불 거래일 경우 필요한 잔액이 있는지 확인한다. 하지만 포크가 일어나는 경우, 이러한 검증 과정만으로 충분하지 않다. 예를 들어 블록 B가 나온 시점에서 김철수씨가 100개의 비트코인을 가지고 있었다고 하자. 블록 C에는 김철수씨가 70개의 비트코인을 이영희씨에게 준다는 거래내역이 기록되어있고, 블록 D에는 50개의 비트코인을 이지은씨에게 준다는 거래내역이 기록되어 있다고 하자. 블록 C와 D를 만든 참여자에게는 잘못이 없다. 블록 B를 기준으로 김철수씨에게는 충분한 잔액이 있었다. 하지만 이 두 거래내역 모두가 실제로 일어날 수는 없다. 이런 문제를 해결하기 위해서 블록 C와 D 둘 중에 한 블록만이 '정격체인'에 편입되고 그 블록에 들어있는 거래내역만이 실제 일어난 거래내역이 된다.

아래의 그림에서 화살표는 해시함수로 만들어진 연결을 가리킨다. 블록 A에서 블록 B로 이어지는 화살표는 블록 A의 해시값이 블록 B에 들어있다는 의미로, 이 해시값은 블록 B를 가지고 있는 사람이 블록 A의 무결성을 확인할 수 있게 해준다. 하지만 블록 B를 가지고 있다고 해서 블록 C와 D중에 어느 쪽이 맞는 블록인지는 확인할 수 없다.

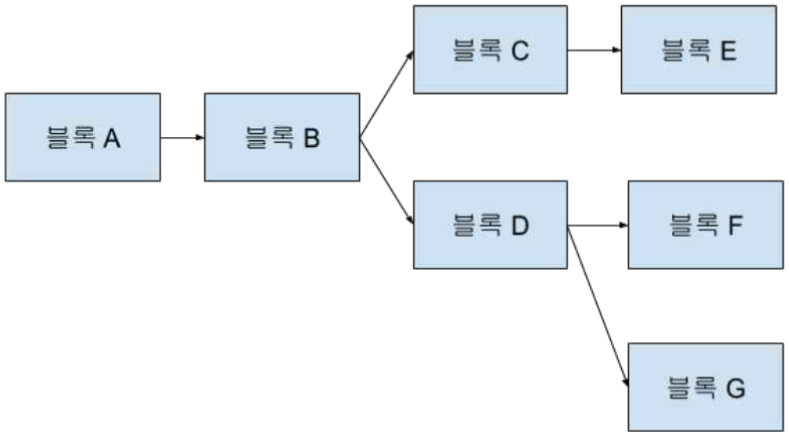


그림 3 포크가 발생한 블록 체인 예

1.3. 포크가 생기는 이유

위에서 보듯 포크는 이중지불같은 문제를 일으킬 수 있다. 이를 막기 위해서, 애초에 포크가 일어나지 않게 할 수는 없을까? 다시 말해서, 위의 그림에서 블록 C와 블록 D처럼 같은 높이 (블록 A로부터 같은 거리)에 두 참여자가 블록을 만드는 일을 피할 수는 없을까? 전혀 오류가 없는 환경이라면 가능하다. 각 높이마다 블록을 생성할 참여자가 자동으로 공평하게 정해지고, 그 참여자가 차별없이 트랜잭션을 최대한 많이 골라 블록을 만들어서 모든 참여자들에게 빠르게 공유해준다면 포크가 생기지 않을 것이다. 하지만 현실은 블록을 만들어서 받는 리워드를 받기 위해 참여자들이 서로 경쟁하는 상황이고, 참여자들 간에 공평하게 블록을 생성할 기회를 줄 권위있는 기관이 존재하지 않고, 새 블록이 모든 참여자들에게

공유되기까지는 예측할 수 없는 시간이 걸린다. 어떤 참여자가 블록을 생성할지 결정하는 방법으로 가장 많이 쓰이고 있는 작업증명과 지분증명의 예를 통해 포크가 생기는 이유를 설명하였다.

1.3.1. 작업증명을 사용하는 경우

작업증명을 사용하는 블록체인의 경우, 작업증명이 동시다발적으로 일어날 수 있기 때문에 여러 블록이 동시에 생성되는 것을 막을 수는 없다. 이 경우 작업의 난이도가 어려워질수록 (예를 들어 비트코인 (Satoshi, 2008) 의 경우 해쉬 퍼즐의 난이도가 올라갈수록) 동시에 생성되는 블록의 갯수가 줄어들겠지만, 난이도가 너무 어려워져서 이 작업을 수행할 수 있는 채굴자가 소수가 되면 중앙화문제가 생긴다. 실제로 지금 비트코인의 경우 경쟁이 심화되어 5개의 채굴업체가 70%의 블록을 생성하고 있다.[4] 그럼에도 불구하고 2개 이상의 채굴업체가 동시에 블록을 생성하는 것을 완전히 막을 수는 없고, 따라서 포크가 일어나는 것을 완전히 막을 수도 없다. 따라서 작업증명을 사용하는 블록체인의 경우, 동시에 생성된 블록들 중에 어느 것이 정격체인에 들어갈지 정하는 분명한 규칙이 필요하고, 이런 규칙이 있어도 일단 모든 채굴자가 생성된 모든 블록을 받은 다음에서야 이 규칙을 적용할 수 있기 때문에 일단 포크가 생기면 정격체인에 들어갈 블록이 정해지기까지는 시간이 걸린다.
이것은 작업증명뿐만 아니라 다른 증명방식을 써도 마찬가지로 일어날 수 있는 문제로, 탈중앙화된 시스템에서는 모두가 "함께" "같은" 결정을 내리기 위해서는 모두가 일단 소통해야하는 전제를 만족시키는 데에 드는 시간을 줄이기 어렵다. 예를 들어 가장 작은 해쉬값을 가진 블록이 정격 블록이 되는 규칙을 만들었다고 해보자. 중국에 있는 채굴자가 블록 C를 만들었고, 미국에 있는 채굴자가 블록 D를 만들었을 때, 프랑스에 있는 채굴자가 블록 C와 블록 D를 둘 다 받아볼 때까지 정격 블록은 정해지지 않는다. 게다가 프랑스에 있는 채굴자가 블록 C를 받은 시점에서, 다른 블록 D를 기다려야한다는 것을 어떻게 알 수 있을까? 인터넷에서 블록이 전송되는 시간은 일정하지 않아서, 얼마나 기다려야 블록 D같은 다른 블록을 모두 받을 수 있을지 정하는 것은 매우 어렵다. 물론 확실하게 하기 위해 매우 오래 기다리는 방법도 있지만, 그럴 경우 블록이 생성되는 시간이 그만큼 길어지게 되고 그로 인하여 거래의 처리 속도가 떨어지게 되며 중앙은행 거액결제에서의 실시간총액결제 (Real-Time Gross Settlement, RTGS) 방식과 같이 실시간 처리에 대한 필요성이 높을수록 문제가 된다.[5]

1.3.2 지분증명을 사용하는 경우

지분증명을 사용하는 경우, 일단 지분을 소유하고 있다는 것을 블록체인에 기록해서 증명하고 나면 여러 가지 방법을 통해 블록 생성에 참여하게 된다. EOS (2018) 처럼 일반 참여자의 투표를 통해서 가장 많은 득표수를 얻은 21개의 노드들이 돌아가면서 블록을 생성하게 되는 경우, 어느 노드가 블록을 생성할 차례인지 분명하게 알려져있기 때문에 포크가 생길 확률이 현저히 낮다. 물론 지금 블록을 생성해야할

---

[4] https://www.coindesk.com/bitcoin-is-becoming-more-decentralized-indicates-new-research

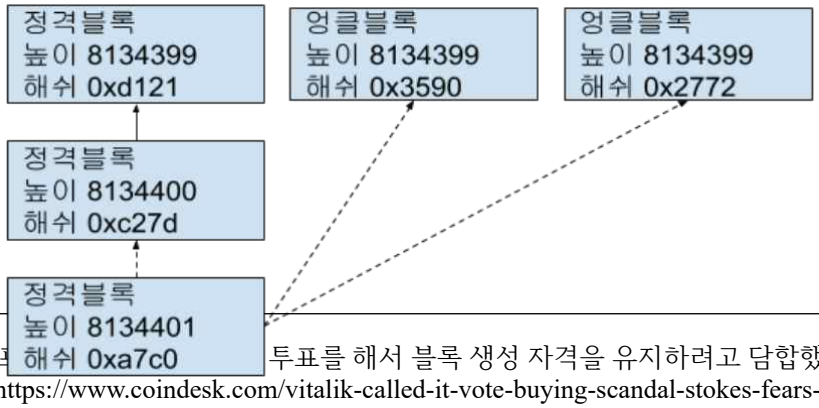
[5]반면, 이연차액결제 (Deferred Net Settlement, DNS) 방식의 경우에는 보다 유용성이 높을 수 있다.

노드가 일시적인 장애를 경험하는 경우, 다음 노드가 만드는 블록과 비슷한 시기에 블록을 만들게 되므로 포크가 생길 가능성이 전혀 없는 것은 아니지만, 정상적인 환경에서 작업증명처럼 2개보다 더 많은 블록이 한꺼번에 생성되기는 매우 어렵고, 따라서 거래의 완결성도 비교적 빨리 확보할 수 있다. 다만 이렇게 블록을 만들어서 더 많은 토큰을 보유하게 된 21개의 노드들이 서로에게 투표하는 방식으로 담합을 조장할 수 있다.[6] 담합이 일어난 시스템에서는 토큰이 일부 참여자들에게 더 편중되기 쉽고, 따라서 탈중앙화를 유지하기 어렵다.

여러 블록체인을 서로 연결하는 역할을 하고자 하는 인터체인 프로젝트 Cosmos (Kwon, 2019)의 경우에도 지분증명을 사용한다. 이 프로젝트의 validator가 되기 위해서는 최소한의 적립금을 맡겨야하고, 맡긴 노드들 중에서 가장 지분이 많은 100개의 노드들이 validator가 된다. Validator들은 가지고 있는 지분에 비례해서 블록을 생성할 기회를 얻게 되는데, 생성한 블록은 정격블록 후보가 되고, PBFT (Practical Byzantine-fault Tolerent) (Castro, 1999) 합의 알고리즘을 거쳐 정격블록이 된다. EOS와 마찬가지로 블록을 생성할 차례인 노드가 장애를 겪고 있는 경우, 다음 차례의 노드까지 2개의 노드가 같은 높이의 블록을 생성할 수 있다. 이런 문제가 생기는 근본적인 원인은 현재 인터넷의 구조적인 한계에 있다. 인터넷은 "best-efforts delivery"라고 해서, 보낸 메세지가 반드시 전달된다는 보장도 없고, 보낸 메세지가 보낸 순서와 다른 순서로 도착할 수도 있다.

1.4. 포크 현황

이더리움 (Ethereum, 2019)에서 포크가 일어나는 경우, 같은 블록 높이에 여러 개의 블록이 생성되고, 그 중 단 한 개의 블록만이 정격블록이 된다. 채굴자들의 입장에서는 자신이 가진 해쉬파워보다 더 많은 해쉬파워를 가진 다른 채굴자가 있을 경우, 채굴에 참여할 동기부여가 어려워진다. 그런데 채굴에 참여하는 채굴자들의 전체 해쉬파워가 적어질수록 시스템의 안전성이 떨어지기 때문에 제일 많은 해쉬파워를 가지지 않았어도 채굴에 참여할 동기를 부여해주기 위해 이더리움은 "엉클 블록"(uncle block)과 "엉클 리워드"(uncle reward)를 도입했다. 엉클 블록은 정격체인에 들어가지 못했지만 정격블록과 유사한 시기에 생성된 블록으로, 포크를 일으킨 블록이라고 할 수도 있다. 블록 내용에 전혀 문제가 없으나 정격블록보다 근소한 시간차로 생성되었거나 채굴자의 네트웍 사정에 따라 늦게 알려진 블록이다. 현재 블록 리워드가 2 ETH이고, 엉클 리워드는 1.6 ETH에 이른다.



[6]실제로 EOS 블록 프로듀서들이 투표를 해서 블록 생성 자격을 유지하려고 담합했다는 주장이 대두된 일이 있다. https://www.coindesk.com/vitalik-called-it-vote-buying-scandal-stokes-fears-of-eos-failure

그림 4 정격블록과 엉클블록

아래 그림은 이더리움 블록 분석 싸이트중 가장 잘 알려진 Etherscan (https://etherscan.io/) 에서 제공하는 엉클 블록 자료로[7], 2018년 1월 10일 하루에만 무려 2,094개의 엉클 블록이 생성된 것을 알 수 있다. 하루에 생성되는 이더리움 블록이 5,760개 정도인 것을 감안할때, 2~3개의 정격 블록마다 1개의 엉클 블록이 생성되었다고 할 수 있다. 2019년 2월에 엉클 블록 숫자가 줄어든 것은 이더리움 버전 업그레이드를 위한 하드 포크 시기와 맞물려있는 것으로 보이고, 하드포크 이후 하루에 500개 정도의 엉클 블록이 지속적으로 생성되고 있는 것을 관찰할 수 있다. 평균 정격 블록 10~11개마다, 2~3분마다 1개의 엉클 블록이 생기고 있다.

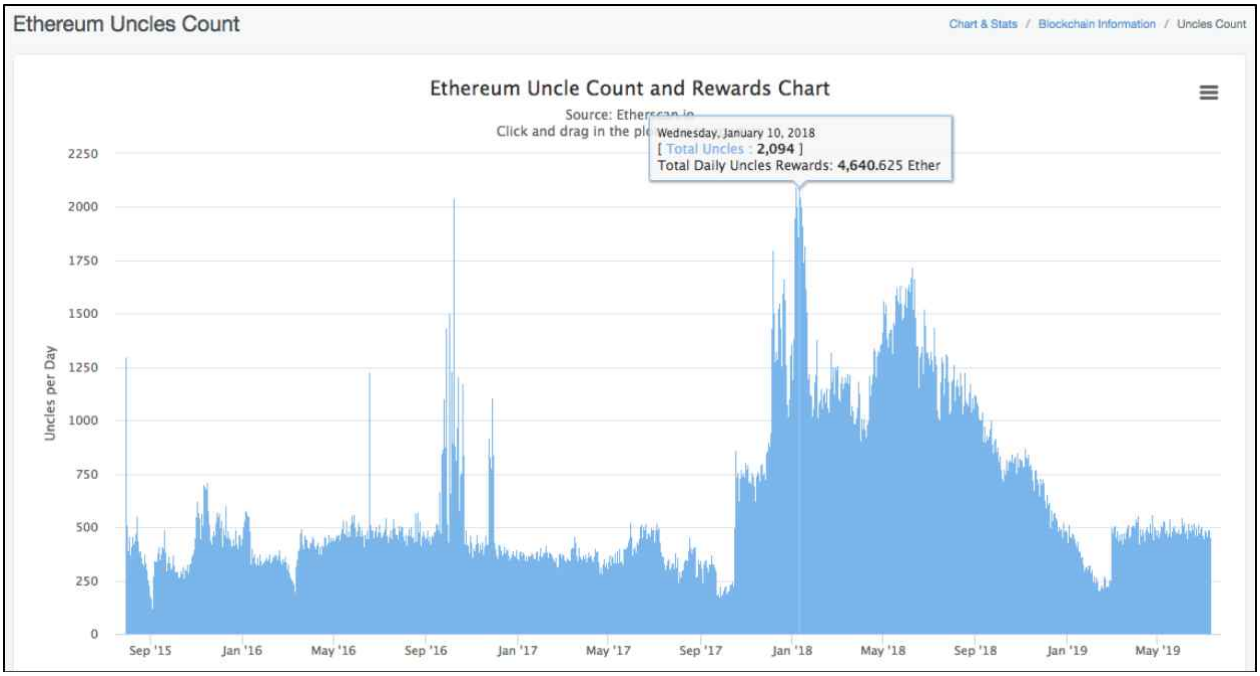


그림 5 이더리움 엉클 블록 생성 현황 예

2. 블록체인과 합의

2.1 합의를 통한 포크 문제의 해결

위에서 언급했듯이 지분증명을 사용해도 현재의 불완전한 인터넷 상황에서 포크가 일어나는 것을 완전히 막을 수는 없다. 포크가 일어났을 때 탈중앙화된 해결을 하기 위해서는 어느 블록을 정격블록으로 정할 것인가에 대하여 블록체인 참가 노드들 사이에 합의가 필요하며, 이 합의에는 다음과 같은 요인들이 고려되어야 한다

[7] https://etherscan.io/chart/uncles

1. 동의 (Agreement) - 모든 정상적인 노드가 같은 블록을 정격블록으로 인식한다.
2. 적합성 (Validity) - 정격블록으로 인식된 블록은 정상적인 노드가 만든 블록이어야한다.[8]
3. 종료 (Termination) - 모든 정상적인 노드가 정격블록을 인식하고 합의 알고리즘을 종료해야한다.

"정상적인" 노드가 생성한 블록이 "모든 정상적인" 노드들에게 정격블록으로 인식되고 안전하게 종료해야 한다는 이 모든 조건을 만족시키는 것은 현재와 같이 불완전한 인터넷 환경에서 생각외로 매우 어려운 문제다.

2.1.1 비잔틴 장군 문제 (Byzantine General Problem)

Lamport (1982)는 비잔틴 장군 문제에서 연락병의 안전이 보장되지 않을 때 두 장군이 동시에 적을 습격하기 위해서 "같은 시간"에 합의하는 것이 얼마나 어려운지 보여준다.[9] 한 장군이 다른 장군에게 '익일 05시'를 제안했다고 해보자. 익일 5시는 점점 다가오고, 연락병에게서는 소식이 없다. 어떻게 해야할까? 이 장군에게는 1) 연락병이 가다가 적에게 잡혀서 다른 장군이 연락을 받지 못한 경우와 2) 다른 장군이 연락병의 연락을 받았지만 익일 5시에 합의하지 않고 연락병을 계속 붙잡고 있는 경우, 그리고 3) 다른 장군이 연락병의 연락을 받고 익일 5시에 공격할 것에 동의했으나 연락병이 돌아오는 길에 적에게 잡힌 경우를 구분할 수 없다. 그러므로 이 장군은 본인이 제안한 익일 05시에 공격을 개시하지 못한다. 마찬가지로, 다른 장군은 익일 5시에 공격할 것에 동의했어도 연락병이 돌아가는 길에 적에게 잡혔을 경우 상대방이 공격하지 않을 것이고, 연락병이 무사히 돌아갔는지 여부를 알 수 없기 때문에 익일 5시에 공격을 개시하지 못한다.
이 이야기는 메세지가 없어질 수 있는 인터넷 환경에 대한 직접적인 비유로, 한 노드는 다른 노드에게 블록 후보를 보냈을 때 1) 블록이 가다가 없어진 경우, 2) 다른 노드는 받았으나 이 노드가 그 사실을 알지 못하는 경우, 그리고 3) 다른 노드로 이 블록을 정격블록으로 쓰기로 합의했으나 이 노드가 모르는 경우를 구분할 수 없다. 말하자면 블록(이 담긴 메세지)가 없어질 수 있는 환경에서 합의는 불가능하다. 이것은 게임 이론에서 내쉬 균형의 전제조건으로 요구되는 "common knowledge problem"의 성립 여부와도 맥을 같이 한다.[10]

2.1.2 비동기 네트워크에서의 불가능 정리

3명의 직장인이 다수결로 점심 메뉴를 결정하기로 했다고 해보자. 각자 다른 층에서 일하는 이 세 사람은 메신저를 이용해서 메뉴를 결정하고 식당에서 만나기로 한다. 1층의 직장인이 된장찌개를

---

[8] 이 정의는 기존의 합의 알고리즘에서의 **유효성**(validity) 정의와는 다르다. 블록체인이 탈중앙화된 시스템이라 모든 honest node들이 같은 블록을 가지고 시작하기 어렵다는 점을 반영한 정의이다.

[9] Gray J. (1978) "Notes on Database Operating Systems." Research Report RJ 2188, IBM

[10] Rubinstein (1989) updated Gray's (1978) version of coordinated attack problem to develop his famous "email" game, which shows that coordinations fail under almost, but not full, common knowledge.

제안하고, 2층의 직장인이 김치찌개를 제안했다고 해보자. 결정권을 가진 3층의 직장인은 답장이 없다. 회의중일까? 상사에게 갑자기 보고중인가? 아니면 전화기가 방전되었나? 언제 이 사람에게 답장이 올지 알 수 없다. 이때 1, 2층의 직장인 두 사람은 언제까지 답장을 기다려야할까?
비동기 네트워크는 메세지가 전달되는 데에 얼마나 시간이 걸릴지 알 수 없는 환경을 말한다. 위의 이야기에서 3층의 직장인이 언제 답장을 보낼지 알 수 없는 것이 비슷한 예인데, 실제로는 답장을 보내도 상대방이 받을 때까지의 시간이 얼마나 걸릴지 알 수 없고, 심지어 그 시간이 무한대인 경우, 그러니까 메세지를 받지 못하는 경우도 생길 수 있다. 문제는 1, 2층의 직장인들에게는 3층의 직장인이 1) 답장을 보내지 않은 경우, 2) 답장을 보냈지만 시간이 아주 오래 걸리는 경우, 3) 답장을 보냈지만 일부에게만 (1, 2층 직장인들 중 한 명에게만) 전달되는 경우들을 구분할 방법이 없기 때문에, 언제까지 기다려야할지 알 수 없다는 데에 있다.
따라서 비동기 네트워크에서는 단 한 명의 참여자만 오류가 날 가능성이 있어도 나머지 정상적인 참여자들이 무한정 기다리는 결과가 나올 수 있기 때문에, 반드시 합의가 일어난다는 보장은 불가능하다. 이것이 유명한 비동기 네트워크에서의 합의 불가능 정리이다. (Fisher, 1985)

2.1.3 동기 네트워크에서의 불가능 정리

3명의 직장인이 다수결로 점심 메뉴를 결정하기로 했다고 해보자. 이번에도 메신저를 이용해서 메뉴를 결정하기로 하는데, 이 메신저는 단체에게 메시지를 보낼 수 없고, 1대 1로만 메시지를 보낼 수 있다고 가정해보자. 1층의 직장인이 된장찌개를 2층과 3층의 직장인에게 제안한다. 2층의 직장인이 김치찌개를 1층과 3층의 직장인에게 제안한다. 이 경우 동기 네트워크이므로 3층의 직장인이 대답하는 데에 걸리는 시간은 제한되어있다. 이때 3층의 직장인이 1층의 직장인에게는 된장찌개를, 2층의 직장인에게는 김치찌개를 먹자고 하면 어떻게 될까? 1층의 직장인은 2표를 얻은 된장찌개가 오늘 점심 메뉴라고 생각할 것이고, 2층의 직장인은 2표를 얻은 김치찌개가 오늘 점심 메뉴라고 생각할 것이다. 합의가 이루어졌다는 오해를 하고 있지만, 실제로 합의는 이루어지지 않았다.
이 3층의 직장인처럼 어떠한 이유에서든 비정상적인 행동을 하고 그로 인하여 발생하는 오류를 비잔틴 (Byzantine) 오류라고 한다. 위 예시처럼 전체 참여자들의 3분의 1 혹은 그 이상이 비잔틴 오류를 보일 경우 합의가 불가능하다.
위에서 메세지가 없어질 수 있는 네트워크와 메세지가 도달하는 시간이 하염없이 늦어질 수 있는 비동기 네트워크에서 합의가 불가능하다는 예시를 보였다. 메세지가 도달하는 데 걸리는 시간이 한정되어있는 동기 네트워크에서는 합의가 가능하지만, 참여자들의 절대 다수 (3분의 2이상) 이 정상적으로 행동할 때만 가능하다.

2.1.4 시빌 공격 (Sybil attack)

어느 웹싸이트에 가입하기 위해서 우리나라에서는 흔히 주민등록번호나 휴대폰번호를 기반으로 한 "본인 인증"을 요구한다. 이렇게 본인 인증을 거쳐야 가입할 수 있는 웹싸이트의 경우, 한 사람이 계정을

하나만 가지고 있게 된다. 하지만 외국의 웹싸이트들은 이메일 주소를 기반으로 하는 경우가 많고, 그런 경우 한 사람이 계정을 여러 개 개설할 수 있다. (사용자 약관에 그런 행동을 부적절하다고 명시하는 웹싸이트들도 많이 있지만 현실적으로 그런 계정들을 일일이 파악하고 삭제하는 것은 쉽지 않다.)
이렇게 다른 환경에서, 웹싸이트에서 투표를 한다고 생각해보자. 민주주의의 원칙에 의거에 한 계정당 한 표를 행사할 수 있다고 할 때, 우리나라의 웹싸이트에서는 한 사람이 한 표를 행사하게 되지만, 외국의 웹싸이트에서는 한 사람이 여러 개의 이메일 주소를 등록하고 여러 표를 행사하는 것도 충분히 가능하다. 이렇게 한 사용자가 여러 사용자가 누려야할 권리를 남용하는 것을 다중인격장애가 있는 소설 주인공의 이름을 따서 씨빌 공격이라고 한다. (Douceur, 2002)
위에서 설명했듯이 합의를 위해서는 동기 네트워크여야하고 전체 참여자중 적어도 3분의 2가 정상적인 행동을 해야한다. 이 조건을 모두 만족시키기 위해서 많은 분산 시스템에서 참여자를 엄격하게 심사하여 고른다. 하지만 블록체인을 기반으로 하는 탈중앙화된 시스템에서는 참여자를 심사할 방법이 없기 때문에, 모든 사람들, 특히 비잔틴 오류를 가진 사람들도 참여할 수 있고, 시빌 공격을 사용하여 오류가 있는 참여자들의 숫자가 실제보다 훨씬 더 크게 시스템에 영향을 미칠 수 있게 되므로, 전체 참여자중 3분의 2 이상이 정상적인 행동을 할 것이라는 보장을 하기가 어렵다.

2.1.5 블록체인에서 사용하는 합의를 위한 전제

시빌 공격이 가능한 비잔틴 오류가 있는 참여자는 블록체인 합의의 천적이다. 이런 참여자가 대부분인 시스템은 제대로 작동할 수 없다. 따라서 블록체인을 기반으로 하는 시스템의 경우, 참여자의 숫자 대신 이런 참여자가 가지고 있는 자원에 제한을 둔다. 작업증명의 경우 전체 해쉬파워중 3분의 1 미만이 이런 비잔틴 오류가 있는 참여자의 관리하에 있다고 가정한다. 지분증명의 경우 전체 지분중 3분의 1 미만이 오류가 있는 참여자의 관리하에 있다고 가정한다. 이런 전제를 가지고 있으면 참여자의 숫자보다는 전체 시스템에 미칠 수 있는 영향력에 제한을 두게 된다. 실제로 이 가정들이 어떻게 사용되는지는 다음 섹션들에서 살펴볼 수 있다.

2.2 나카모토 합의

윗 장에서는 분산 시스템, 특히 탈중앙화된 시스템에서 합의가 얼마나 어려운지에 대해서 설명했다. 또한 현재의 불완전한 인터넷 환경에서는 블록 생성 속도를 어느 정도 일정하게 유지하면서 포크를 완전히 없애는 것은 불가능하다. 그렇다면 실제로 블록체인에서는 어떻게 포크가 일어났을 때 어느 블록이 정격블록이 될지 합의할까? 현재 가장 많은 사용자와 사용량을 보이는 비트코인과 이더리움 모두 나카모토 합의(Nakamoto Consensus)를 사용한다.

2.2.1 나카모토 합의란 무엇인가

나카모토 합의는 참여자들이 어느 블록이 정격블록이 되어야할지 투표를 하는 방법으로 결정하되, "자원을 소비하는" 방식의 공개투표를 하게 되어있다. 예를 들어, 쓰레기종량제의 종량제봉투를 사야 쓰레기를 버릴 수 있듯이, 투표용지를 돈을 주고 사야한다고 생각해보자. 내가 지지하는 후보가 당선되게 하기 위해서 나는 돈을 지불하고 한꺼번에 여러 장의 투표용지를 살 수 있다. 하지만 투표용지를 사기 위해서 돈을 내는 것이 아니고 운동장을 한 바퀴 뛸 때마다 한 표를 받을 수 있다고 해보자. 운동장을 한 바퀴 뛰는 데에 걸리는 시간은 사람마다 다르겠지만, 아무리 빠른 사람이라도 일정 시간동안 한 표 이상을 얻을 수는 없다. 나카모토 합의는 이렇게 자원을 소비하는 방법으로 한꺼번에 여러장의 표를 사기 어렵게 만든 다음, 가장 표를 많이 받은 체인을 정격체인으로 간주한다.

나카모토 합의에서 투표는 어떻게 이루어질까? 나카모토 합의에서 투표는 우리가 흔히 생각하는 1인 1표, 전원이 투표에 참가하는 선거와 다르다. 참여자들은 자기가 정격블록이라고 생각하는 블록의 해쉬값을 자기 블록에 포함해서, 자기가 정격블록이라고 생각하는 그 블록의 child block을 생성한다. 위 그림 3에서 블록 C와 블록 D가 생성되었을때, 블록 E를 만든 참여자는 블록 C가 정격블록이라고 투표한 것이다. 마찬가지로, 블록 F와 블록 G를 만든 참여자들은 블록 D가 정격블록이라고 투표했다.

비트코인이나 이더리움의 경우, '운동장을 뛰는' 대신 각 참여자들은 어려운 해쉬퍼즐을 풀고, 그 해쉬퍼즐이 어려울수록 자원을 더 많이 사용해야하고 따라서 더 많은 표를 행사할 수 있다. 각 블록마다 해쉬퍼즐의 난이도가 기록되어 있고, 그 난이도가 높을수록 많은 자원을 사용해서 해쉬퍼즐을 풀었다고 추정할 수 있기 때문에 난이도의 합이 가장 높은 체인이 정격체인이 된다. 예를 들어서 블록 A, B, C, D, E, F 모두 난이도가 10이라고 하고, 블록 G의 난이도가 20이라고 해보자. 체인 A-B-C-E와 A-B-D-F는 총 난이도 합이 40이고, A-B-D-G는 50이 되기 때문에 정격체인은 50표를 받은 A-B-D-G가 된다.

나카모토 합의를 이용하는 비트코인과 이더리움에서 정격체인은 '커뮤니티에서 가장 표를 많이 받은 체인'이라고도 알려져 있고, '가장 긴 체인'이라고도 알려져 있다. 이는 같은 높이에 있는 블록들은 최저 난이도가 정해져있고, 최저 난이도 이상으로 어려운 해쉬퍼즐에 도전하는 경우 다른 블록이 먼저 생성되어 정격블록이 될 가능성이 높기 때문에 같은 높이의 블록은 비슷한 난이도를 갖기 때문이다. 예를 들어, 위의 그림에서 블록 A, B, C, D, E, F는 모두 최저 난이도 10을 만족시켰다고 하고, 블록 G는 난이도 20을 만족시켰다고 하자. 블록 G를 만드는 데에 드는 시간이 블록 E, F를 만드는 시간보다 훨씬 오래 걸리기 때문에 다른 참여자들은 블록 E나 F에서 연결되는 블록을 만들 가능성이 높아진다. 따라서 긴 체인일수록 더 길어질 가능성이 높기 때문에, 가장 긴 체인이 가장 높은 난이도의 합을 가지게 되는 경우가 많아서 '가장 긴 체인이 정격체인'이라는 말이 대부분 참이 된다.

2.2.2 나카모토 합의의 장점

나카모토 합의의 가장 큰 장점 중에 하나는 아무도 전체 참여자 명단을 확보할 필요가 없다는 것이다. 탈중앙화된 시스템에서는 언제 어디서 새로운 참여자가 나타날지 알 수 없고, 따라서 전체 참여자 명단을 확보할 방법이 없다. 또한 아무도 참여자를 심사하지 않기 때문에, 참여자의 "본인 확인"등은 불가능하다. 누구나 익명으로 댓글을 남길 수 있는 게시판을 생각해보자. 이 게시판에서 절대다수가 게시판 관리자를 퇴출시킬 것을 요구한다고 해보자. 실제로 이 댓글을 올린 사람은 과연 몇 명일까? 익명성을 포기하지 않으면서 한 사람이 올릴 수 있는 댓글 숫자를 제한하는 방법으로, 위에 예시로 든 "운동장 한 바퀴마다

한 표"처럼 "운동장 한 바퀴마다 댓글 하나" 라고 한다면, 과연 한 명이 몇 개의 댓글을 올릴 수 있을까? (운동장 한 바퀴를 뛰었다는 것은 쉽게 확인할 수 있고 위조하기 어렵다고 가정하자.) 나카모토 합의는 각 참여자가 일정량 이상의 자원을 소비한 다음에만 투표할 (댓글을 달) 수 있게 함으로써 씨빌 공격을 완전히 뿌리뽑지는 못하더라도 그 심각성을 낮출 수 있다.
우리가 흔히 하는 모든 선거에서 투표일 이전에 "선거인단 명부"를 작성해야한다. 누가 투표를 할 자격이 있는지, 자격이 있는 사람은 몇 명이나 되는지, 이 사람들이 투표를 할 수 있게 마련해야 할 투표소의 규모는 어떻게 되는지, 실제로 투표를 하고 집표를 하기 이전에 일어나야할 많은 일들을 나카모토 합의에서는 생략함으로써 비용을 절약한다.

2.3 투표를 통한 합의

그렇다면 차후에 강력한 해쉬파워를 가진 공격자가 들어와도 과거에 일어난 결제내역을 변경할 수 없게 하려면 어떻게 정격체인을 결정해야할까? Agreement, Validity, 그리고 Termination 모두를 만족하는 기존의 합의 알고리즘 연구 결과를 블록체인에 적용하면 어떨까? 비잔틴 오류가 있는 참여자까지 허용할 수 있는 합의 알고리즘은 이미 많이 알려져있다. 대표적인 것으로 PBFT (Practical Byzantine Fault-Tolerant) 합의 알고리즘 (Castro, 1999) 이 있고, Ripple (Chase, 2018) 등의 블록체인 시스템들이 이 알고리즘을 기반으로 한 합의 알고리즘을 사용하고 있다.
기존의 비잔틴 오류를 허용하는 합의 알고리즘에는 크게 두 가지가 있다.

1. 리더가 합의할 내용을 결정한다. 나머지 참여자들은 리더를 감시하다가 리더가 이상한 행동을 보일 경우, 나머지 참여자들이 소통하여 리더를 변경한다.
2. 모든 노드가 합의할 내용을 제안하고, 서로 소통해서 그 중 한 개의 값으로 합의한다.

EOS의 경우 1과 유사한 방식으로 작동한다고 할 수 있다. 21개의 블록 프로듀서들이 돌아가면서 리더 역할을 맡아 블록을 생성하고 홍보한다. 나머지 참여자들은 (꼭 블록 프로듀서가 아니어도 가능) 이 블록 프로듀서들을 관찰하다가 이상한 행동을 보일 경우 다음 기회에 이 블록 프로듀서에게 투표하지 않는 방식으로 리더를 변경할 수 있다. 반면, 알고랜드 (Gilad, 2017) 는 2와 유사한 방식으로 작동한다고 할 수 있다. 알고랜드는 매 블록마다 합의에 참여할 노드들이 무작위로 선출되고, 그 노드들간의 통신을 통해 한 개의 값으로 합의한다.
이 두 종류의 합의 알고리즘 모두, 참여자들간의 소통을 통해 시스템이 오류가 있더라도 그것을 극복하고 꾸준히 기능하도록 만들어준다. 이때 소통이란 사실 투표로 기능한다. 예를 들어, 리더가 이상한 행동을 보였다고 소통하는 것은, 리더를 경질하자는 불신임 투표를 한 것과 마찬가지다. 반대로 현 리더를 다음 리더로 추천하는 것은, 리더를 유지하겠다는 지지 투표를 한 것과 마찬가지다. 다시 말해서, 비잔틴 오류를 허용하는 합의 알고리즘들은 대체로 투표를 기반으로 하고 있다.
투표를 기반으로 한 합의 알고리즘의 장점은 탈중앙화를 잘 지원한다는 것이다. 투표한 값을 블록체인에 기록해두면 나중에 들어오는 참여자도 손쉽게 현재의 시스템 상황을 파악하고 합의 알고리즘에 참여할 수 있다. 사용하는 증명방식에 따라 (e.g. 작업증명, 지분증명) 조금씩 다르지만, 많은 참여자들이 투표에 참여할 수 있어서 좀더 민주적이라고도 할 수 있다.

반대로 단점은 오프라인 선거에서 봤던 모든 문제점들이 그대로 반복된다는 것이다. 오프라인 선거에서 볼 수 있었던 담합과 표 사고 팔기 등이 블록체인에서도 그대로 나타나게 된다.
탈중앙화된 합의는 투표를 통해 이루어지게 되며, 가령 투표 방식에 단순 최다득표제(plurality)가 적용된다면 합의의 내용은 가장 많은 표를 받은 블록을 정격 블록으로 정하게 되는 것이다. 적절한 합의 제도의 설계와 구현을 위해서는 다음과 같은 기술적 요인들이 고려되어야 한다.

- 어디에 투표할 것인가 (정격블록, 정격체인/완결성 등)
- 누가 투표할 수 있는가?
- 투표의 유효성은 어떻게 확인할 수 있는가?
- 투표는 언제 시작하고 끝내야 하는가?

또한 다음과 같은 경제적 요인들이 고려되어야 한다.

- 어떤 방식(plurality? Simple-majority? Supermajority? Borda~~Vorda~~ rule? Condorcet rule?)
- 투표에 대한 동기는 어떻게 제공해야 하는가?
- 적정 수준의 보상은 어떻게 정해야 하는가?
- 어떻게 "빈익빈 부익부"를 통제할 것인가?

2.4 이더리움 네트워크에서의 합의 구현

2.4.1 공학적인 관점에서 알려진 이더리움 1.0의 문제점들

현행 이더리움(1.0)에서의 합의 알고리듬은 위에서 기술한 작업증명을 기반으로 한 나카모토 합의 알고리듬이다. 비트코인이 결제만을 위한 블록체인 시스템인 반면, 이더리움은 solidity 프로그래밍 언어로 구현한 분산앱(decentralized App, dApp)을 실행시킬 수 있는 범용 플랫폼을 구현하였고, 많은 사용자를 확보하여 블록체인과 암호화폐의 확산에 많은 공을 세웠다. 하지만 사용자가 늘어나면서 문제점도 발견되었다.

완결성(finality) 결여 문제

비트코인과 이더리움 1.0에서 사용하는 나카모토 합의는 엄격한 의미에서의 합의는 아니다. 위에서 말한 합의의 조건인 동의 (Agreement), 적합성 (Validity), 종료 (Termination) 3가지 중 적합성은 만족시키지만 동의도 네트웍 상태에 따라 시간이 걸릴 수 있고, 무엇보다 종료가 언제 일어나는지 알기 어렵다. 위에 제시한 게시판 예시를 다시 생각해보면, 게시판에 어느 정도 댓글이 달리면, 혹은 어느 정도 시간이 지나면 관리자가 퇴출될지 그렇지 않을지 결정할 수 있을까? 게시판에 100개의 댓글이 달렸고, 그 중 30개의 댓글만 관리자 퇴출을 요구한다면 관리자는 연임되어도 될 것 같다. 하지만 나중에 관리자 퇴출을 요구하는 110개의 댓글이 올라온다면, 관리자는 그때 퇴출될 수도 있을 것이다. 여기에 관리자 연임을 요구하는 댓글 300개가 다시 올라온다면, 관리자를 다시 데려오면 될까?

이 문제가 결제망에서는 결제의 완결성과 직결된다. 비트코인이나 이더리움 1.0에서 정격블록으로 통상 인정하는 것은 '가장 긴 체인'에서 최소한 6개의 블록이 그 뒤로 연결되는 경우이다. 하지만 6개의 블록이 그 뒤로 이어졌다고 해서 과연 이 블록이 정격블록이라고 확신할 수 있을까?

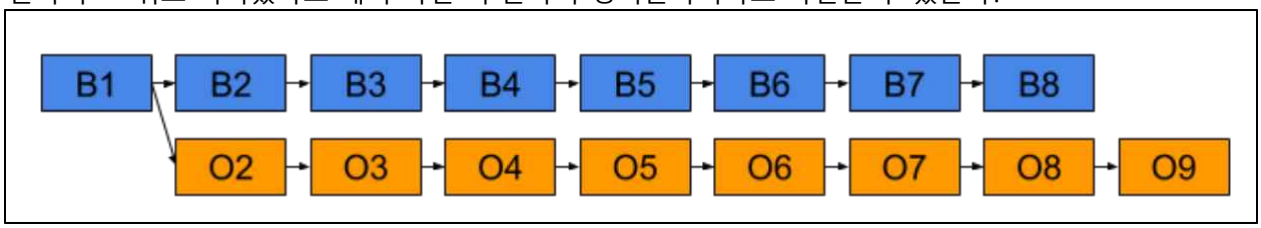


그림 6 최장 체인 방식의 정격 체인 확정시 문제점

위의 그림6에서 파란색 블록들이 참여자들에게 알려져있고, 오렌지색 블록들은 특정 참여자가 만들어서 아무에게도 알리지 않았다고 가정해보자. 다른 참여자들은 파란색 블록들만 알고 있으므로 B1과 B2는 정격블록으로 인정할 것이다. B2 블록안에 어느 사용자가 거래소에 암호화폐를 판매하는 거래내용이 포함되어 있다고 하자. 거래소는 B2를 정격블록이라고 간주하므로 암호화폐에 해당하는 기축통화 (원화, 달러화 등등)을 사용자에게 지불한다. 하지만 오렌지색 블록들이 거래소를 포함한 다른 참여자들에게 알려지는 순간, B2 블록에서 사용자가 거래소에 판매했던 내용은 무효화되고 암호화폐는 사용자의 계정으로 돌아간다. 실제로 비트코인과 유사하게 나카모토 합의를 사용하는 비트코인골드의 경우 2018년 10월 강력한 해쉬파워를 가진 공격자가 비트코인골드를 현금화한다음 정격체인을 갈아엎는 방법으로 같은 비트코인골드 화폐로 거래소의 현금을 여러 차례 인출하는 공격을 성공시킨 바 있다. [11] 나카모토 합의는 참여자들간에 블록 생성전에 협의할 필요가 없고, 모든 참여자가 협력할 경우 높은 확률로 완결성을 제공할 수 있어서 비트코인과 이더리움의 성공에 상당히 기여했다는 평가를 받고 있다. 하지만 비트코인과 이더리움의 성공은 합리적이지만 이기적인 참여자들도 불러와서, 이제 나카모토 합의만으로는 안전한 결제환경을 제공하기 어려워졌다

처리속도문제 (확장성의 결여)

2017년 11월 크립토키티라는 첫 이더리움 기반 게임이 출시되어 큰 인기를 얻으면서, 트랜잭션 숫자가 급증하였다. 문제는 이더리움의 블록 생성 속도는 비교적 일정하게 정해져 있어서, 급증한 트랜잭션 숫자를 감당할 수 없었다는 점이다. 점점 오래 기다려야하는 상황이 되자, 사용자들은 점점 더 많은 트랜잭션 fee를 지불해야했고, 고양이 하나를 새로 교배시키는 데에 15만5000달러가량이 들기도 했다. [12]

흔히 특정 웹싸이트나 어플리케이션에 사용자가 폭주하면 더 용량이 큰 서버 혹은 더 많은 서버를 투입해서 해결하지만, 비트코인이나 이더리움은 더 많은 채굴자 (dApp을 실행시키는 노드들) 가 참여한다고 해서 트랜잭션 처리속도가 빨라지지 않는다. 비트코인은 2017년말에 SegWit [13] 이라는

---

[11] Fortune, 'Bitcoin Spinoff Hacked in Rare '51% Attack', May 2018. (https://fortune.com/2018/05/29/bitcoin-gold-hack/)

[12] https://medium.com/cryptokitties/cryptokitties-birthing-fees-increases-in-order-to-accommodate-demand-acc314fcadf5

[13] https://en.wikipedia.org/wiki/SegWit

프로토콜 변형을 도입했는데, 이것은 한 블록에 들어가는 트랜잭션의 갯수를 거의 2배까지 늘릴 수 있는 개선안이었지만, 트랜잭션 숫자가 그보다 더 늘어날 경우 유연하게 처리속도를 늘릴 수는 없다.

중앙화 문제

2019년 6월 시점에 단 2개의 채굴 풀 (Ethermine과 SparkPool)이 이더리움 전체 생태계중 52%의 해쉬파워 (연산능력)를 가지고 있다는 것이 알려졌다.[14] Ethermine이 28%, SparkPool이 24%를 각각 가지고 있어서, 다른 채굴 풀에 비해 작업증명을 더 빠르게 완수할 수 있다. 이더리움에서 과반수 이상의 정격블록이 이 두 개의 채굴 풀에 의해 만들어진다. 실제로 이더스캔 웹싸이트에서 제공하는 아래 그림에서 2019년 6월 28일부터 7월 10일까지 만들어진 블록들을 보면 아래의 그림과 같이 두 채굴 풀에서 만들어진 블록들이 50%를 조금 넘는다.

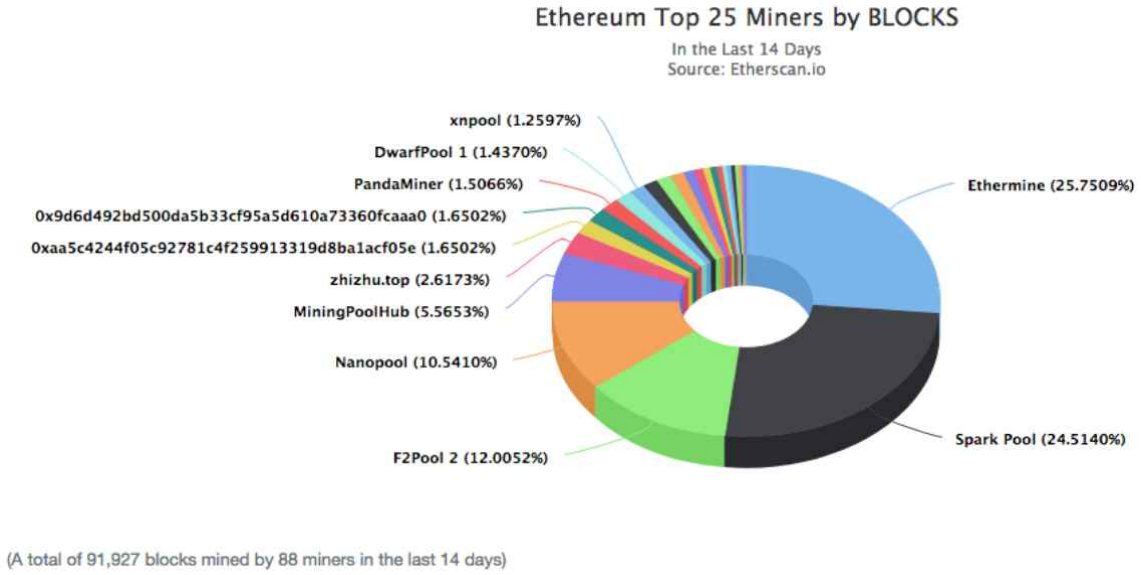


그림 7 이더리움 채굴 현황

이렇게 소수의 채굴 풀들이 많은 블록을 생성하고 그 리워드를 받아가는 것은 빈익빈 부익부 현상을 가중시킨다. 소수의 채굴 풀들이 계속해서 리워드를 받아 설비에 재투자할 수 있는 반면, 나머지 채굴 풀들은 채굴에 들어간 비용을 회수하지 못할 수도 있다. 또한 이렇게 소수의 채굴 풀이 많은 블록을 생성할 경우, 특정 트랜잭션을 정격체인에 빨리 넣고 싶거나 혹은 넣지 않고 싶을 때, 이 채굴 풀들에게 블록체인 밖에서의 소득을 제공하는 "뇌물 공격"이 가능해진다. 또한 이 채굴 풀들이 담합하는 경우, 정격체인을 바꿀 수도 있다. 완결성을 논의하면서 예시로 사용했던 파란색 블록들과 오렌지색 블록들을 생각해보자. 파란색 블록들이 정격체인이었을 때 채굴 풀들이 담합해서 오렌지색 블록들을 더

[14] https://coinnounce.com/ethereum-is-centralized-2-mining-pools-control-more-than-50-hashrate/

생성하기로 하면, 오렌지색 블록들이 더 많아져서 정격체인이 변경될 수도 있다. 따라서 중앙화 문제는 빈익빈부익부뿐만 아니라 시스템의 안전성에 영향을 미친다.

2.4.2 이더리움 2.0 - 지분증명 (Proof of Stake)

이더리움 2.0는 "가장 어려운 해쉬 퍼즐을 푼" 가장 긴 체인을 고르는 대신 좀더 직접적인 투표를 통해 커뮤니티의 의견을 반영하여 포크를 해결하고 정격블록을 정한다. 이더리움 1.0 알고리듬에서 아래 포크가 일어난 블록들을 보면, "가장 긴 체인"이라는 조건에 가장 잘 부합하는 것은 초록색 블록들이다. 하지만 하얀색 블록들이 계속 늘어날 경우 정격체인은 하얀색 블록들로 바뀔 것이다. 이렇게 완결성이 보장되지 않고, 실제로 정격체인을 바꾸는 공격이 다른 작업증명 기반의 암호화폐에서 이미 일어난데다, 중앙화 문제와 작업증명에 쓰이는 전기 등등의 자원 낭비가 환경에 좋지 않다는 비난을 피할 수 없다는 여러 단점이 제기되어 이더리움 2.0은 지분증명을 사용하고 LMD GHOST (Ethereum, 2019a)와 Casper FFG(Ethereum, 2019b)를 혼합한 알고리즘으로 포크를 해결한다. LMD GHOST와 Casper FFG의 보다 자세한 구현 내용은 부록 1에서 조금 더 자세히 기술한다.
이더리움 2.0에서 블록을 생성하기 위해서는 일정량의 (현재 제안된 양은 32) ETH를 예치금(deposit)으로 지불해야한다. 현재 제안된 양은 32 ETH, 2019년 7월 10일 시세로 약 미화 9천불, 한화 1천만원 정도에 해당하는 금액이다. 예치금 영수증이 블록체인에 기록되면 이것이 지분 증명으로 인정되어 PoS 활성 검증 집단(active validator set)을 저장하고 관리하는 비컨(beacon) 체인에 참여할 자격이 주어진다. 비컨 체인에 참여하는 노드들은 (임의로) 돌아가면서 차례대로 블록을 제안하고, 나머지 노드들은 그 블록이 제대로 만들어졌는지 확인할 의무를 갖는다. 이 모든 의무들을 제대로 수행하는 노드들은 예치금(stake)에 비례한 이자를 받고, 블록에 들어가있는 트랜잭션들 수수료(fee)도 일부 받을 수 있다.

2.5 경제학적 관점에서의 문제점들

구성원들이 복수의 대안들 중 하나를 선택을 하는 방법은 여러가지가 있지만 이는 결국 투표(voting)라는 형태를 갖게 된다. 현재 가장 일반적으로 이용되는 투표 방법은 단순 최다득표 승리(plurality) 방식이지만, PBFT(Practical Byzantine Fault-Tolerance) 알고리듬을 이용하는 이더리움 계열 블록체인의 경우 정격 블록 및 체인에 대하여 투표하고자 할 때는 종종 2/3 초과반수(supermajority)를 요구하기도 한다.
만일, 3개 이상의 대안들을 놓고 투표를 통해 하나를 선택해야 할 경우 투표 문제는 보다 복잡해진다. 단순 최다득표 외에도 가령 콘도세 방식(Condorcet method)라고 알려진 모든 대안들을 대상으로 양자택일 투표를 실시해 최종적으로 남은 대안을 선택하는 방식을 택할 수 있고, 보다(Borda) 방식이라고 알려진 모든 대안들에 대하여 투표자가 선호에 따른 순위를 부과하는(rank-choice) 방식을 택할 수 있으며, 그 외에도 점수를 대안들에 대하여 차등 부과하는 방식 (e.g. Copeland Index), 양자 대결 상황에 이를 때까지 복수 회차(round)를 적용하는 등 세부적인 적용 사항들을 변경함에 따라 다양한 투표 방법을 만들어낼 수 있다.

투표 방법이 중요한 이유는 그에 따라 선택되는 대안이 바뀔 수 있기 때문이다. 가령 2010년 미 캘리포니아 주 오클랜드 시장 선거에서는 복수 라운드 선호투표(Multi-round ranked-choice) 방식을 도입했는데, 이때 시장으로 당선된 Jean Quah는 마지막 양자 대결 라운드 직전까지도 단순 최다 득표로는 2위에 머물렀으며 일반적인 단순 최다득표자 승리 방식이 적용되었다면 시장에 당선되지 못했을 것이다.[15] 따라서, 이를 인지하고 있는 사람들은 투표에서 자신의 선호를 정직하게 드러내는 대신 전략적으로 행동할 수 있다.

Arrow(1957, 2014)는 어떠한 방식을 택하더라도 "일반적으로" 투표자들의 선호를 왜곡 없이 종합해내어 사회적으로 최적의 결과를 도출할 수 없다는 점을 증명하였다 (Impossibility Theorem). 즉 어떠한 방식을 통해서도 개별적으로 볼 때 타당한 주요 원칙(axiom)들을 만족시키며 개별 투표자의 선호가 정직하게 드러나고 사회적으로 최적인 결과를 얻을 수는 없다는 것이다.[16] Arrow는 (불)가능성 정리에서 조작(불)가능성을 만족시켜야 할 원칙이라고 명시적으로 기술하지는 않았지만, 이 문제는 불가능성 정리와도 밀접한 관계가 있는 동시에 투표 방식 설계에서 매우 중요한 문제이다. Gibbard(1973)와 Satterswaite(1975)는 Arrow의 불가능성 정리와 유사한 맥락에서, 3개 이상의 대안을 놓고 사회가 합의에 이르고자 할 때, 조작불가능한 동시에 독재적이지 않은 합의 방식은 존재하지 않는다는 것을 각각 증명하였다.

위의 정리들은 블록체인에서 투표를 이용하여 합의를 구현하는 것에 대해 부정적인 함의를 담고 있다고 볼 수 있지만, Arrow 및 Gibbard-Satterswaite 정리는 서수적 선호(ordinal preference)만을 이용하여 사회적 선택을 구현할 경우에 대하여 증명한 결과이다. 그런데, Saari (2000) 등은 기수적(cardinal) 특성을 이용할 수 있을 경우 부정적 함의를 벗어날 수 있는 가능성을 제시한다. 실제로 블록체인은 참가자에게 금전적, 즉 기수적 보상을 제공하는 구조이며, 이는 지분증명 방식에서 각 노드가 갖는 정격 블록 후보들에 대한 선호 차이의 '정도'(degree)를 알 수 있다면 이를 보상의 정도와 연동할 수 있다는 뜻이다. 또한, 지분증명 방식에서 정격블록 선정을 위한 투표 설계를 살펴보면 실제로 사회적 선택 이론에서 고려하는 외부 조건에 대한 전제들과는 차이가 존재함을 알 수 있다. 우선, 각 노드가 정격블록에 대해 투표할 때에는 최장(longest) 혹은 최중(heaviest) 체인 등 주어진 규칙(rule)에 따를 것이 권고되며, 그렇지 않을 경우 불이익이 가해진다는 점에서 이더리움 2.0 등 지분증명 방식에서 채택하는 투표는 Arrow의 (불가능성) 정리에서 제시된 "외부적인 규칙의 영향이 없이 사회적 선택이 이루어져야 한다"는 원칙을 지키지 않고 있다고 볼 수 있다. 또한, 특정 관리자 노드가 투표 어젠더 조작(agenda manipulation)을 이용한 합의 왜곡을 시도할 경우, 조작의 수단으로 쓰이는 스포일러(spoiler) 후보의 도입이 필요한다. 그런데, 블록체인의 경우 작업증명이나 지분증명 모두 시빌 공격을 막기 위해 신규 노드 개설과 예치금 혹은 작업수행 등에 있어서 적지 않은 비용을 요구하는 구조를 갖추고 있으며, 이로 인해 관리자 노드는 조작에 대한 동기를 잃을 가능성이 높다. 결국 지분증명에서의 투표는 사회적 선택을 이끌어내는 수단보다도 탈중앙화된 Peer-to-peer 네트워크에서 생길 수 있는 정보의 전파(propagation) 지연 및 손실에 대한 보완 방법이라고 이해하는 것이 적절할 것이다.

---

[15] https://www.acgov.org/rov/rcv/results2010-11-02/rcvresults_2984.htm

[16] 보다 자세한 설명은 부록 2에 기술되어 있다.

2.5.1 조작(manipulation) 및 정보 캐스케이드(cascade)

블록체인 환경과 같이 모든 행동이 기록되고 관찰 가능한 상황에서 투표를 구현할 때 추가로 고려해야 하는 문제는 정보 캐스케이드(information cascade) 효과이다. 정보 캐스케이드는 모든 참가자에게 드러난 공공 정보(public information)가 자신만이 알고 있는 개인 정보(private information)보다 신뢰도가 높다고 판단되는 경우, 자신의 정보를 무시하고 공공 정보에 따라 판단을 내리게 되며, 그로 인하여 새로운 정보가 사회적으로 축적되지 않으면서 모든 참가자들의 선택이 동일해지는 쏠림(herd) 현상이 일어나는 상황을 지칭한다.[17] 이 경우 효율적인 정보의 통합(aggregation)이 이루어지지 않게 되면서 사회적으로 효율적이지 않은 대안을 선택하게 되는 상황이 발생할 수 있다 (Bikhchandani, et al., 1998 등).

그런데, Easley and Kleinberg (2010)는 조작가능성(manipulability)과 개방된 환경에서의 순차투표 구조로 인한 정보 캐스케이드는 사실상 동일한 성격의 문제가 될 수 있음을 지적한다. 다음 예를 통해 살펴보자. 네트워크 전체에 두 개의 정격 블록 후보 (b = 0, 1)가 있다. 정해진 기간동안 발생한 모든 블록 생성 기록이 모든 노드들에게 다 전파되어 알려진다면 이 중 하나의 후보만이 규칙에 따라 정격 블록으로 인정받게 된다. 네트워크에 아무런 데이터 지연 및 유실 문제가 없을 때의 정격 블록을 1이지만, 네트워크 사정으로 인하여 어떤 노드는 후보 0을, 어떤 노드는 후보 1를 정격으로 인지하고 있으며, 아무런 정보가 없을 경우 각 후보가 정격 블록이 될 확률은 0.5(라플라시안 원리)라고 하자. 과반수의 확률(P > 1/2)로 각 노드는 정격 블록을 정확하게 인식하고 있으며, 편의상 정격 블록 인식은 독립적으로 이루어진다고 가정하자.[18] 노드의 수가 적당히 많다면 모든 노드를 대상으로 동시에 비밀 투표가 진행할 경우 높은 확률로 블록 1이 정격 블록으로 선택될 것이다.

지분증명 기반 검증에서 모든 투표 기록은 블록체인에 저장되고 증명되어야 한다. 그런데, 각 참가자(validator)는 투표가 진행되는 동안 자신보다 일찍 투표한 다른 참가자들—네트워크 지연으로 인하여 전부 다는 아니더라도—이 어떤 블록에 투표했는지 볼 수 있다. 만일 첫번째로 투표한 노드가 블록 0을 선택했다고 하자 ($b_1 = 0$). 이는 첫번째 노드의 사적 신호(private signal) $s_1 = 0$임을 의미한다. 다른 노드들은 s_1을 관찰할 수 없지만 투표 기록 b_1은 관찰할 수 있다. 따라서 두번째로 투표하는 노드는 첫번째 노드의 사적 신호가 $s_1 = 0$이라는 것을 알 수 있다.

만일 $s_2 = 0$이라면 $Pr(b = 0 \mid s_1 = 0, s_2 = 0) = 2P - P^2 > 3/4$을 만족하게 된다. 따라서 두번째 노드도 블록 0을 선택한다 ($b_2 = 0$). 그런데, 만일 $s_2 = 1$이라면 $Pr(b = 0 \mid s_1 = 0, s_2 = 1) = 1/2$이 된다. 즉, 이 경우 두번째 노드는 후보 0과 1에 대해 무차별하게 된다. 이 경우 두번째 노드는 1/2의 확률로 후보 0과 1 중에 선택한다고 하자.

세번째 노드는 첫번째 노드가 $s_1 = 0$이라는 것은 알 수 있으며, 두번째 노드의 투표 기록이 $b_2 = 1$일 경우에는 $s_2 = 1$이라는 것도 알 수 있다. 하지만, $b_2 = 0$일 경우 $s_2 = 0$ 이기 때문에 선택한 것인지 $s_2 = 1$이고 확률 1/2로 둘 중 하나를 선택하는 과정에서 $b_2 = 0$을 택했는지 알 수 없다. 다만 전자($s_2 = 0$)일 가능성이 더 높다는 것은 알고 있으며, 결국 $s_3 = 1$인 경우라 하더라도 이를 무시하고 $b_3 = 0$을 택한다.

---

[17] 그러나 그 역(Herd가 발생하면 information cascading이 일어난다)이 항상 참인 것은 아니다.

[18] 실제로는 네트워크 구조에 영향을 받을 가능성이 높다.

이 경우 실제로는 s_2 =1, s_3 = 1 이며, 전체 노드들에게 주어진 정보를 취합한다면 후보 1이 시스템상의 정격 블록으로 보다 적합하다고 결론을 내려야 함에도 불구하고 b_2 = 0, b_3 = 0이 됨을 알 수 있다. 이후의 n번째 노드들도 세번째 노드 이후의 노드들의 투표에서 얻을 수 있는 새로운 정보가 아무것도 없다는 사실을 알고 있기 때문에, 첫번째 노드와 두번째 노드의 투표기록에서 유추 가능한 정보에 따라 세번째 노드와 같은 선택, 즉 b_n = 0을 택한다.
결과적으로, 정보 캐스케이딩이 일어나기 시작한다면 각 노드가 자신이 갖고 있는 정격 블록에 대한 정보가 앞서 이루어진 투표 기록에 따른 과반수 노드와 다름에도 불구하고 (초)다수결에 따른 정격노드 합의에서 벗어난 블록을 선택해서 생기는 불이익을 피하기 위하여 자신의 정보와 달리 기존에 선택된 다른 블록에 투표할 수 있다는 것이다. 물론, 이 경우 지분증명 알고리듬 구현 방식에 따라 페널티가 부과될 수 있다. 그러나, 가령 GHOST 의 heaviest chain 규칙 위반에서 오는 페널티의 크기보다 Casper the FFG에서 합의에 성공한 노드에 투표함으로써 얻는 보상이 크다면 이는 개별 노드 입장에서는 합리적 선택이 된다.
특히, Casper와 같이 지분 크기에 따라 투표의 비중이 결정되는 경우라면 이 문제는 더욱 심각해질 수 있다. 지분이 적은 노드는 가능한 한 다른 노드의 결정을 두고 보면서 자신이 소수에 투표하게 됨에 따라 블록 보상을 가능성을 줄여야 할 동기가 있는 반면, 지분이 큰 노드는 자신이 먼저 움직임으로써 적은 지분의 노드들의 선택을 자신의 선택과 같아지도록 유도할 수 있기 때문이다 (Gul and Lundholm, 1995 등). 만일, 이를 피하기 위해 외생적으로 투표 순서를 부과할 경우 합의에 걸리는 시간이 길어진다.
결과적으로, 지분증명 알고리듬 구현에서 투표 순서를 결정하는 것은 사전적/외생적일 경우에도, 내생적일 경우에도 모두 문제가 존재한다. 그러나, 만일 이 문제를 피하기 위해 비대면 분산 네트워크에서 비공개 투표를 수행함으로써 사전에 수행된 투표에 대한 기록을 참가자들이 알지 못하게 한다면 투표의 취합 및 결과 도출 과정에서 위변조와 관련된 신뢰의 문제가 새롭게 발생하게 된다.

2.5.2 노드 간의 부익부, 빈익빈 심화

대개의 지분증명 시스템에서 보상은 지분에 비례해서 주어진다. 각 노드의 운영 비용이 크게 차이가 나지 않는다고 가정하자. 이는 PoW를 대신하여 PoS가 등장한 배경을 고려한다면 타당한 가정이다. 노드 운영 비용을 제외하고, 노드가 받은 보상을 지분으로 재투자하는 경우, 원금에 대해 복리 이자가 붙듯이 이제 노드 사이의 지분과 보상을 더한 값의 격차가 점점 벌어지게 된다.
노드 n의 보상을 구해보자. K는 외생적으로 주어지는 보상의 크기, m은 운영비용, a_n은 현 시점에서 보유한 지분의 크기, A_n은 현 시점에서의 지분률이며, 이 갖는 블록의 정격 인증에 따른 순 보상을 r이라고 하자. 이는 다음 식으로 나타난다.

$$r_n = KA_n - m$$

여기서 $A_n = \frac{a_n}{\sum_{1=1}^{N} a_i}$ 이다. 만일, 모든 노드가 자신의 순 보상을 모두 자신의 지분을 늘리는데 이용한다면 노드 n의 다음 단계 지분율은 $A'_n = \frac{a_n + r_n}{\sum_{1=1}^{N} a_i + r_i}$과 같이 나타나며, 이 때 수익 r'_n은

$$r'_n = KA'_n - m = K\frac{a_n + r_n}{\sum_{1=1}^{N} a_i + r_i} - m$$

과 같이 나타낼 수 있다. 이는, 다음과 같이 다시 쓸 수 있다.

$$r'_n = K\frac{a_n + KA_n - m}{\sum_{1=1}^{N} a_i + K - Nm} - m = KA'_n - m$$

즉, 부등식 $A'_n = \frac{a_n + kA_n - m}{\sum_{1=1}^{N} a_i + K - Nm} > A_n = \frac{a_n}{\sum_{1=1}^{N} a_i}$ 을 만족할 경우, 다음 단계의 순 보상이 증가한다.
이는 다음 부등식으로 정리된다.

$$A_n N > 1$$

결과적으로 지분 증명 방식을 운용할 때 지분에 비례해 블록 보상이 이루어지며, 지분을 늘리는데 아무런 제약이 존재하지 않는다면, N개의 노드가 존재할 때 지분율이 1/N보다 높은 노드들은 계속 수익을 얻고, 그렇지 않은 노드들은 점차 수익이 낮아지면서 순 보상이 0 미만이 되는 노드들이 떨어져 나가고, 신규 노드의 진입 장벽은 점차 높아지게 되면서 집중화가 가속될 것이다. 물론, 기술적으로는 노드들의 지분 증가분에 대해서 제약을 거는 방식 등을 적용하여 집중화를 막을 수는 있지만, 초기 참여자에 대한 강한 인센티브를 제공하려는 대부분의 신규 개방형 블록체인 시스템에서 처음부터 쉽게 채택하기는 어려우며, 결국 이에 대한 합의는 알고리듬이 아닌 해당 블록체인 시스템의 거버넌스(governance)에 달려있다고 볼 수 있다.

3. 합의의 구현 방향 및 활용 가능성

실제로 구현된 모든 사회제도와 마찬가지로 블록체인에서의 합의 조건이 Arrow의 정리에서 제시하는 "일반적" 원칙을 모두 만족시켜야 하는 것은 아니며, 그럴 수도 없다. Maskin (2016)은 조작가능성이 "완전히" 없어야 한다는 요구는 과도하며, 보다 적절한 질문은 "어떤 투표 방식이 가장 조작가능성을 낮출 수 있을 것인가?" 라고 하였다.[19] 본 장에서는 앞서 설명한 이더리움 2.0에서 제시된 합의

[19] 그는 IIA(Independence of Irrelevant Alternatives) 조건을 완화할 경우 콘도세(Condorcet) 규칙과 보다(Borda) 규칙을 조합한 다수결 원칙을 적용하면 바람직한 결과를 얻을 수 있다는 의견을 제시하였다.

알고리듬과 블록의 특징들에 기초하여 합의의 구현 방향 및 중앙은행의 결제 업무에의 활용 가능성에 대하여 살펴본다.

3.1 지분에 따른 내생적 투표 순서 선택 동기 부여

현재 대부분의 지분증명 알고리듬에는 지분 다수결(majority stakes) 및 지분 비중에 따른 선형 보상(linear compensation)방식을 채택하고 있다. 그런데, 지분 비중이 높은 노드의 경우는 다른 노드들보다 먼저 투표를 수행하고 자신의 투표 결과를 지분이 낮은 노드들이 관찰하여 따라가게 만들 동기를 갖게 되는 반면, 지분이 낮은 노드는 자신의 정격 블록 선정 관련 정보와 무관하게 다른 노드들의 투표를 기다렸다가 이를 그대로 따라갈 동기를 갖게 된다. 만일, 지분 크기 외에 노드의 운영비용이 별 차이가 없다면 노드간 지분 격차가 벌어지게 됨을 확인하였다. 이를 종합하여 고려한다면 향후의 보상 체계가 보완해야 할 부분이 정보 손실로 인하여 최적이 아닌 정격 노드 선택이 일어나는 경우를 최소화하고 부익부, 빈익빈을 완화해야 함을 짐작할 수 있다. [20]

우선, 지분이 낮은 노드들이 투표에 보다 먼저 적극적으로 임할 동기를 부여할 필요가 있다. 이는 지분이 낮은 노드들이 먼저 투표할 경우 정격노드로 선정되지 않은 노드에 투표할 경우라도 나중에 투표할 경우보다 페널티의 크기를 적게 부과하는 방식 등을 고려할 수 있을 것이다. 반면 노드의 지분이 증가함에 따라 나중에는 이와는 반대 방향의 동기를 부과할 수 있을 것이다. 동일한 기간(epoch) 및 동일한 후보 노드들을 대상으로 한 투표라 하더라도 지분의 크기 및 투표의 순서에 따라 페널티와 보상의 정도가 달라지기 때문에 개별 노드 입장에서는 투표 타이밍에 대한 인센티브가 달라지게 되며, 이는 정보 캐스케이드를 완화하는 효과를 가져오게 된다. 이에 대한 scoring rule 적용 구현 예를 살펴보자.

두 노드 1과 2가 있다. 시스템과 노드는 정격블록 가능성에 대하여 모두 동일한 사전적 믿음을 갖고 있다. 노드 i's (i = 1, 2)의 payoff는 다음 식과 같다고 하자.

$$u(t_i, z_i) = -\alpha(s_i, t_i) f(z_i - w) + \beta(t_i) + \gamma(s_i)$$

w : 확정 정격 블록

s_i : 노드 i의 stake 비중

z_i : 노드 i가 선택한 후보 블록

t_i : 노드 i의 투표 시간 ($t_i \in (0,T)$)

α, β: 두번 미분 가능하고 연속이며 양인 t_i의 함수

f: punishment 함수 (quadratic, log, sgn 등 적절한 함수를 선택)

γ : 지분 보상 함수($\gamma' > 0$) (지분이 큰 쪽이 더 많은 기대 효용을 갖도록 보정하는 수단으로 이용)

만일, $\alpha(t)$, $\beta(t)$가 다음 특성을 만족한다고 하자

---

[20] 실제로 미 해군 군사 법원에서는 정보 캐스케이드를 줄이기 위하여 재판 내에서 증인들의 증언의 순서를 낮은 계급에서 높은 계급으로 정하도록 규정한 것으로 알려져있다. (Bikhchandani et al, 1998)

i. $\partial\alpha/\partial s > 0$ (지분이 높을수록 틀렸을 때의 벌금(penality)이 큼)

ii. $\partial\beta/\partial t < 0$ (나중에 투표할수록 보상(reward)의 크기가 적음)

iii. $\partial^2\alpha/\partial s\partial t < 0$ (single crossing property)

iv. $\beta''/\beta' < \alpha''/\alpha'$

이제 지분이 작은 노드들은 틀렸을 때에 대한 punishment 가 작은 반면 먼저 투표할 때의 보상이 크고, 지분이 큰 노드들은 그 반대가 되면서, 정보 캐스케이드 문제를 회피할 수단을 구현할 수 있다.[21]

3.2 블록체인의 금융 결제망 활용 검토

앞서 제시된 노드 지분에 따른 투표 순서 내생화(endogenization) 알고리듬은 지분증명 방식 블록체인에서 비효율적 신규 블록 생성이 일어날 가능성을 줄여줄 수 있음을 확인했다. 하지만, 금융결제망에 블록체인이 이용되려면 속도 및 용량 측면에서 다음과 같은 요구 사항들을 만족시켜야 한다. 우선 단위 시간당 거래 처리 용량(transactions per second, TPS)이 현재 등장한 블록체인들보다 높아야 하고, 거래 완결성(finality) 또한 짧은 시간 내에 보장되어야 하며, 그 과정에서 적법한 거래가 누락되는 일을 최소화해야 한다. BOK Wire 등과 같이 기존에 구축된 전용망 기반 결제시스템에 비해 경제적으로 효율적이라는 것이 확인되어야 하며, 기존 시스템으로부터 블록체인 방식으로 전환 과정에서 발생할 수 있는 금전적, 심리적 전환비용 또한 무시할 수 없는 고려요소이다.

본 절에서는 경제적 비용이 기존 결제망과 유사하다는 가정 하에서 지분증명(PoS) 방식 블록체인의 금융 결제망 활용 가능 여부에 대해서 개방형(Permission-less) 및 폐쇄형(consortium)의 경우를 모두 살펴본다. BIS (2019)는 작업증명(PoW) 방식 블록체인의 경우, 거래확정과 관련된 과도한 비용 발생 및 수수료 관련 무임승차(free-riding)으로 인한 시스템 안정성 저하 등의 문제로 인하여 금융 결제망 적용 가능성에 대해 부정적인 의견을 피력하였다. 하지만, 지분증명 방식에서는 작업증명 방식을 채택했을 때 발생하는 과도한 에너지 소요 및 비용 문제가 크게 완화된다. BIS가 제기한 수수료 무임승차 문제는 초기 투자자에 대한 인센티브 제공에서 발생한 문제로 보는게 보다 타당하며, 거래 요청자가 일으키는 거래 비용(연산비용 + 저장비용)을 내부화 할 수 있는 알고리듬과 보상체계를 도입함으로써 일정 정도는 해결할 수 있다. 최장체인 방식 대신 최중체인 방식을 도입할 경우 RTGS처리에 적합하지 않았던 긴 블록 생성 주기를 상당히 단축시킬 수 있다. 특히, 블록체인의 개방성 및 탈중앙화 특성을 일정정도 포기하고 노드로 참가할 수 있는 자격을 제한하는 폐쇄형(consortium) 방식을 택할 경우 블록 보상이나 암호화폐 발행 및 관리 등의 문제가 간단해지면서 위의 문제들은 더욱 완화될 수 있다.

그럼에도 불구하고, 본 연구에서는 지분증명방식의 (탈중앙화) 블록체인의 결제망 이용의 타당성 대해서는 현 시점에서는 부정적이라는 의견을 제시한다. 우선, 본 연구에서 주로 살펴본 이더리움2.0과 같이 높은 수준의 개방성을 제공하는 지분증명방식 블록체인을 결제망으로 이용하는 경우를 살펴보자. 인증 노드로 참여하고자 하는 개인 혹은 단체에 일정 정도 이상의 지분을 요구하는 것은 당연하지만, 결제망으로서 운용될 만큼의 성능 및 안정성을 유지하려면 지분과 더불어 블록체인 노드가 제공해야

[21] 스코어링에 대한 보다 자세한 설명은Gneiting et al. (2007)을 참조하라

하는 연산 능력, 저장 용량, 망 안정성 및 속도 등에 대한 고려도 뒤따라야 한다. 즉, 최소 지분(minimum stake)과 더불어 최소 용량(minimum capacity)에 대한 요구 및 확인이 필수적이다. 그렇지 않을 경우, 인증블록으로 선정된 노드가 거래 처리 및 신규 블록 생성 과정에서 처리속도(response time) 및 용량(throughput, transactions per second)이 결제망에서 필요한 수준 이하로 떨어지는 일이 발생할 수 있다. 특히, 데이터 전파 과정에서 지연(delay)에 대한 우려가 큰 공공 네트워크에서 운영되는 블록체인의 경우 거래 완결성 제공에 전용망을 이용할 경우보다 시간이 필요하며 그 와중에 적법한 거래가 한참 시간이 지나서 누락될 수 있다.[22] 이를 막기 위해 지분 비중이 상승할 수록 컴퓨팅 파워를 보다 높게 요구할 수 있지만, 컴퓨터 자원의 성능 및 용량을 비대면 상황에서 검증하는 것은 사실상 불가능한 것으로 알려져 있다.[23] 처리 속도나 용량 측면에서 문제를 일으키는 노드들의 지분을 삭감(slash)함으로써 컴퓨터 자원에 대한 투자 동기를 부여할 수는 있지만, 이는 거래 처리가 소수 노드로 집중되는 현상을 부채질하게 된다. 이 경우, 중앙은행이나 정부의 통제 바깥에 있을지도 모르는 소수 노드들이 금융 결제망을 운용을 하는 형태로 이어질 위험을 감수해야 되기 때문에 제도의 안정성(가령 비경제적 원인에 인한 노드 운용 중지 가능성 등) 측면에서 바람직하다고 보기는 어렵다.

EOS와 같이 인증 노드 자격에 높은 제한(선거를 통해 선출 등)을 두고 감시 노드만 개방적으로 운용하면서 탈중앙화 정도를 약화시킨 블록체인, 더 나아가 아예 허가(permission) 또는 컨소시움(consortium) 형태로 은행이나 금융업, 혹은 금융전산을 담당하고 있는 기업들을 중심으로 보다 폐쇄적으로 블록체인을 이용한 결제망을 구축하는 경우를 생각할 수 있다. 허가형의 경우는 사전적으로 비잔틴(악성) 노드를 걸러냈다고 볼 수 있으며 악성노드가 없거나 매우 적은 비중이라고 볼 수 있다. 시빌 공격에 대한 우려를 크게 덜 수 있기 때문에, 보다 적은 수의 노드를 이용하기 때문에 커뮤니케이션 횟수와 대기시간을 줄이면서 거래를 처리하고 신규 블록을 생성할 수 있다.

그러나, 집단조기인출(뱅크런)이 발생할 수도 있는 금융위기 상황의 경우 개별 기관은 유동성을 확보해야 할 필요성이 평소보다 현저하게 상승한다. 개별 금융 기관들은 자신이 보유한 유동성(reserve)보다 몇 배나 큰 규모의 유동성 흐름(flow)을 다루고 있기 때문에 늘 유동성 수준을 관리하고자 하는 동기를 갖게 된다. 그런데, 금융위기 혹은 그에 준하는 상황이 닥칠 경우 개별 기관은 유동성 위험에 빠지는 상황을 막기 위해 자신의 유동성 수준을 떨어뜨리는 신규 블록 거래처리 결과에 반대표를 던질 수 있다. 그런데, 이러한 상황은 결제망에 가입된 금융기관(노드)들 사이에서 전체 유동성을 놓고 제로섬(zero-sum) 게임을 한다고 볼 수 있으며, 이 때부터는 합의에 도달하는 것이 어려워지게 된다.

우선, 거래처리 완료를 위해 임의로 선정된 단독 노드에 의해 신규 블록이 생성(거래가 처리)되며 일반적인 블록체인의 경우처럼 초과반수(super-majority) —가령 전체 노드의 2/3이상— 동의가 필요하다고 가정하자. 이 경우, 신규 블록이 생성됨에 따라 자신의 순 유동성 감소 폭이 과도하다고 판단하는 노드가 1/3을 넘는다면 신규 블록 생성에 실패하고 관련거래는 완료되지 못할 것이다. 전체 지분 중 2/3이상이 동의하는 거래처리 결과를 도출하기도 어렵지만, 2/3의 노드 (혹은 지분) 이상

---

[22]처리 속도를 높이기 위하여 샤딩(sharding)등 서브체인을 이용한 단계별 처리 방안이 제안되었지만 아직은 구현되지 않았다.

[23] 이 부분은 UC Berkeley 컴퓨터 공학과 교수 Dawn Song으로부터 확인했다.

동의가 성립되는 상황은 실제로는 지분 총합 1/3 이하의 소수 기관의 유동성 문제를 오히려 악화시키는 상황으로 귀결될 가능성이 높게 된다. 특히, 이 시점에서 신규 블록을 생성하게 된 노드는 평상시에는 감수하지 않을 수준의 페널티를 감수하고서라도 자신의 순 유동성이 감소하는 것을 막고자 하는 동기를 가질 수 있다. 특히, 처리할 거래의 내용을 취사선택하는 것이 가능할 경우, 이를 통해 다른 금융기관의 유동성을 사실상 약탈해올 시도를 할 가능성도 존재한다.

초과반수 대신 과반수로 합의 요건을 낮출 수도 있지만 이는 대형 금융기관끼리의 담합이 상대적으로 용이해질 가능성이 증가한다. 신규 블록 생성을 복수의 노드가 수행하고 그 결과를 비교하여 다를 경우 투표를 하는 식으로 운영할 수도 있지만, 이 경우는 개별 블록생성 노드들이 자신에게 유리한 거래들을 취사 선택할 경우 신규 블록 후보의 내용 불일치로 인한 거래 처리 합의 실패 가능성이 증가하게 된다. 만일, 합의 실패에 대해 노드(금융기관)에 부과되는 페널티가 유동성 유지에 대한 지불의사보다 낮다면 노드들은 기꺼이 합의 실패를 선택할 용의가 있다고 볼 수 있다.

4. 정리 및 제언

본 연구는 블록체인에서 투표를 이용한 지분증명(Proof-of-Stake, PoS) 방식 합의 알고리듬에 대하여 이더리움(Ethereum) 2.0에서 제안된 내용들을 중심으로 살펴보고, 결제망에서의 활용 가능성에 대하여 살펴보았다. 정격 블록을 선정하기 위한 과정에서 사회적 선택 이론(social choice theory)에서 알려진 의제 조작, 콘도세 패러독스 등 여러 문제들이 존재할 가능성이 있지만, 오직 대안 사이의 선호 순서만을 고려하는 서수적 선호(ordinal preference) 대신 선호의 크기 차이 또한 고려되는 기수적 선호(cardinal preference)를 반영할 경우 이를 피할 수 있으며, 특히 블록체인의 경우 암호화폐 보상 체계와 규칙 기반 투표를 이용하면 이 문제들을 회피할 수 있음을 확인하였다.

그러나, 현재 이더리움 2.0에서 거래 완결성을 위해 제안된 Casper the Friendly Finality Gadget 알고리듬은 정보 캐스케이딩(information cascading) 문제에 취약할 수 있을 확인하였다. 이 경우, 지분이 큰 노드들의 선행 투표는 지분이 작은 노드들의 선택을 의도적, 혹은 비 의도적으로 적절하지 않은 방향으로 유도할 수 있으며, 결과적으로 전체 노드가 동시에 투표할 경우에 비해 비효율적인 선택이 가능함을 확인했다. 이에 대한 대안으로, 지분율 크기에 따라 투표 타이밍에 대한 인센티브를 제공하는 방식을 제안하였다.

본 연구는 지분증명 (Proof-of-Stake, PoS) 방식 공개(permission-less) 블록체인의 경우에도 작업증명(Proof-of-Work, PoW) 방식 체인과 마찬가지로 결제누락 가능성, 처리 시간, 처리 용량 등의 측면에서 결제망 적용에 어려움이 존재한다고 본다. 금융기관이 노드로서 역할을 하는 폐쇄적 PoS방식 블록체인을 이용한 결제망의 경우는 평상시에는 문제가 없을 수 있지만, 현 알고리듬이 그대로 적용된다면 위기시에는 신규 블록 생성 자격을 얻는 노드가 자신의 유동성을 유지하기 위해 페널티를 감수하고서라도 일부 거래는 누락하거나, 자신에게 유동성이 유입될 때까지 서로 합의를 미룬 상태가 지속되면서 거래처리가 지연되고, 합의가 되더라도 소수의 금융기업에 유동성 부족 문제를 집중시키는 문제가 발생할 수 있다. 따라서, 현 시점에서는 중앙은행등 지급결제망의 관리자가 현재와 같이 독점(혹은 과점)적으로 관리하면서 법적인 의무와 책임을 부여하는 것이 보다 적절할 것으로 판단한다.

부록 1. 이더리움 2.0 지분증명 알고리듬

LMD GHOST (Latest Message Driven Greedy Heaviest Observed SubTree)[24]

이더리움 1.0에서 '가장 긴 체인'은 커뮤니티가 해쉬파워를 이용해 가장 많이 투표한 체인이어야하는데, 실제로는 '가장 긴 체인'이 모든 참여자에게 알려지게 하는 것에 의도하지 않은 (인터넷 장애) 혹은 의도한 (이기적 채굴) 어려움이 있어서 '가장 긴 체인'으로 정격체인을 사용하는 문제점을 (완결성 결여, 공격 가능) 위에서 기술하였다. 이더리움 2.0에서는 '가장 긴 체인' 대신 '가장 무거운 체인'을 정격체인으로 사용한다.

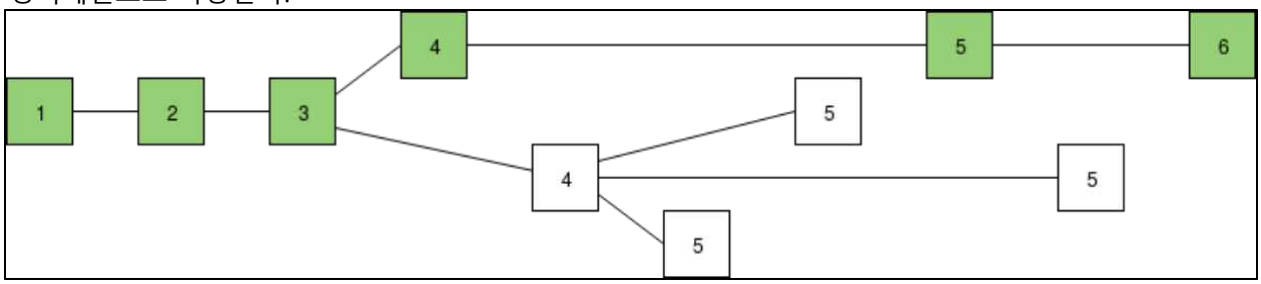


그림 10 최장 체인(longest chain)

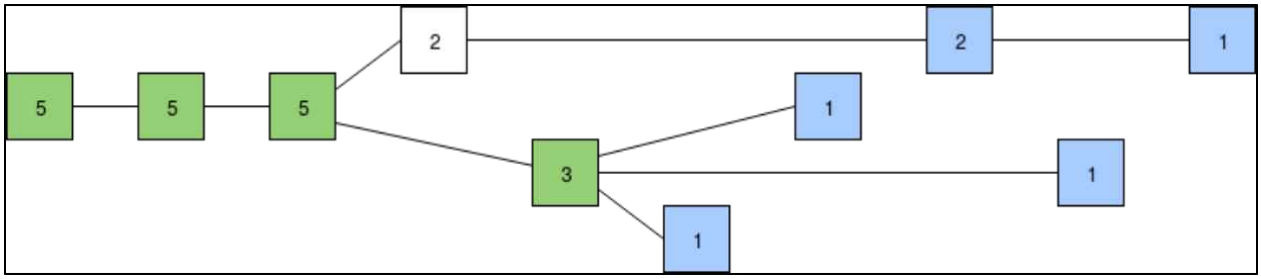


그림 11 최중 체인(heaviest chain)

무게는 얼마나 많은 사용자가 이 블록을 정격블록으로 인정하는 투표 (attestation)를 했는지에 따라 결정된다. 위의 그림에서 각 블록의 숫자들은 이 블록의 득표수와 이 블록에서 연장된 블록의 득표수를 합한 것으로, 참여자들은 가장 최근의 블록으로부터 6개의 블록을 거슬러 올라가 투표할 수 있다. 이렇게 최근의 투표 메세지를 집계하여 가장 표를 많이 받은 체인을 정격체인으로 간주하면 초록색 블록들이 정격체인으로 확정된다. 이 방식의 장점은 이전에 정격체인으로 간주되던 것을 이후 큰 해쉬파워를 가진 채굴자가 혹은 채굴 풀이 손쉽게 뒤집을 수 있다는 약점을 극복한다는 것이다.

위의 그림 11과 같이 초록색 블록들을 정격체인의 블록이라고 간주했을 때, 2표를 받은 하얀색 블록이 정격체인에 들어가기 위해서는 (그리고 3표를 받은 초록색 블록이 정격체인이 아니라고 하기 위해서는)

---

[24] https://vitalik.ca/general/2018/12/05/cbc_casper.html

하얀색 블록이 더 많은 표를 받아야한다. 하지만 포크가 일어났을 때 같은 높이의 두 블록에 모두 투표하는 것은 불법적인 행위로 간주되어 예치금을 일부 삭감당하게 되므로, 정상적인 참여자는 하얀색 블록과 초록색 블록 모두에 투표하지 않는다. 따라서 전체 참여자가 5개의 노드라고 가정했을 때, 초록색 블록에 3개의 표가 주어졌고 하얀색 블록에 2개의 표가 주어진 시점에서, 이 5개의 표가 5개의 서로 다른 노드에서 왔다면 초록색 블록이 정격블록임이 확정되고 변경될 가능성이 존재하지 않는다.

Casper the FFG (Friendly Finality Gadget)

LMD GHOST를 사용해도, 정격블록이 확정되지 않을 때가 있을 수 있다. 위의 그림에서 참여자 5명 중 한 명이 비잔틴 오류가 있어서, 예치금을 잃을 것임에도 불구하고 하얀색과 초록색 블록 둘 다에게 투표했다고 가정해보자. 그런 경우 첫째 이 표를 무효화해야하고, 그 투표자의 예치금을 삭감해야한다. 무효화하고 난 다음 실질적으로 하얀색은 1표, 초록색은 2표를 받은 상태가 된다. 남은 1명이 하얀색에 투표하는 경우 정격블록을 확정할 수 없다. 이러한 문제를 다루기 위해 주기적으로 완결성(finality)을 더해줄 Casper the Friendly Finality Gadget (FFG)을 LMD GHOST에 추가하였다. Casper FFG는 LMD GHOST에서 어떻게 투표할지 규칙을 정해주고, 여기서 결정된 정격체인은 절대로 변경되지 않는다.
Casper FFG는 매 100개의 블록마다 LMD GHOST에서 일어난 attestation을 이용해서 checkpoint를 결정한다. 전체 투표자들 (작업증명과 달리 투표를 할 수 있는 참여자 목록이 모두에게 알려져있다.) 중 3분의 2이상이 투표한 checkpoint는 정당화(justified)되고, 정당화된 checkpoint뒤에 또 다른 정당화된 checkpoint가 체인으로 이어지면 기존 checkpoint는 완결된다. 이때 같은 높이의 서로 다른 checkpoint에 투표하는 경우 예치금을 삭감하고, 기존의 정당화된 checkpoint와 체인으로 이어지지 않는 checkpoint에 투표하는 경우에도 예치금을 삭감하여 초과반수가 같은 checkpoint에 투표하도록 권장한다.
지분증명, LMD GHOST, Casper FFG를 결합하여 이더리움 2.0은 구성원들이 투표를 통해 정격블록을 결정하고, 구성원의 3분의 2 이상이 투표한 정격블록에 대해서는 완결성을 보장한다.

참고문헌

Arrow, Kenneth J. "The origins of the impossibility theorem." The Arrow Impossibility Theorem (2014): 143-148.

Bikhchandani, S., Hirshleifer, D., & Welch, I. (1998). Learning from the behavior of others: Conformity, fads, and informational cascades. Journal of economic perspectives, 12(3), 151-170.

BIS (2019) “Beyond the doomsday economics of “proof-of-work” in cryptocurrencies”

Budish, E. (2018) The Economic Limits of Bitcoin and the Blockchain, Working Paper, University of Chicago

Buterin, V. (2017). Caspter the Friendly Finality Gadget. https://arxiv.org/abs/1710.09437

Castro, M. & Liskov, N. (1999). Practical Byzantine-Fault Tolerance. In the Proceedings of the 3rd Symposium of Operating Systems Design and Implementation (OSDI). Usenix.

Chase, B., &MacBrough, E. (2018) Analysis of the XRP Ledger Consensus Protocol. https://arxiv.org/abs/1802.07242

Douceur, M. (2002). The Sybil Attack. In the Proceedings of International Workshop on Peer-to-Peer Systems (IPTPS 2002,) 251-260.

Easley, D., and J. Kleinberg (2010). Networks, crowds, and markets. Vol. 8. Cambridge university press

EOS (2018). EOS.IO Technical White Paper v2. https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md

Ethereum (2019a). Ethereum 2.0 Phase 0 – Beacon Chain Fork Choice https://github.com/ethereum/eth2.0-specs/blob/dev/specs/core/0_fork-choice.md

Ethereum (2019b). Ethereum 2.0 Phase 1 – Shard Data Chains. https://github.com/ethereum/eth2.0-specs/blob/dev/specs/core/1_shard-data-chains.md

Ethereum (2019c). Ethereum Whitepaper https://github.com/ethereum/wiki/wiki/White-Paper

Fisher, M., Lynch, N., & Paterson, M. (1985) Impossibility of distributed consensus with one faulty process. Journal of the ACM, 32(2), 374-382

Gibbard, A. (1973) "Manipulation of voting schemes: a general result." Econometrica 41, no. 4: 587-601.

Gilad, Y., Hemo, R., Micali, S., Vlachos, G., & Zeldovich, N. (2017) Algorand: Scaling Byzantine Agreements for Cryptocurrencies. https://eprint.iacr.org/2017/454

Gneiting, Tilmann, and Adrian E. Raftery (2007). "Strictly proper scoring rules, prediction, and estimation." Journal of the American Statistical Association 102, no. 477: 359-378

Gray, J. (1981). "The transaction concept: Virtues and limitations." In VLDB, vol. 81, pp. 144-154.

Kwon, J., & Buchman, E. (2019) A Network of Distributed Ledgers (Cosmos) https://cosmos.network/resources/whitepaper

Lamport, L., Shostak, R., & Pease, R. (1982) The Byzantine General Problem. ACM Transactions on Programming Languages and Systems (TOPLAS), 4(3), 382-401

Larimer, D. (2017) DPOS Consensus Algorithm - The Missing White Paper. https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper

Saleh, F. (2021). “Blockchain without waste: Proof-of-stake.” Review of financial studies, 34(3), 1156-1190.

Satoshi, N. (2008) Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf

Satterthwaite, M. A. (1975). "Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions." Journal of economic theory, 10(2), 187-217.