ETRI

2018
Electronics and
Telecommunications
Trends

4차 산업혁명 사회의 초연결 지능과 신뢰 인터넷 기술 특집



# 블록체인과 합의 알고리즘

Blockchain and Consensus Algorithm

**임종철** (J.C. Yim, hektor@etri.re.kr) 네트워크연구본부 책임연구원
**유현경** (H.K. Yoo, hkyoo@etri.re.kr) 네트워크연구본부 책임연구원
**곽지영** (J.Y. Kwak, jiyoung@etri.re.kr) 네트워크연구본부 선임연구원
**김선미** (S.M. Kim, kimsunme@etri.re.kr) 네트워크연구본부 책임연구원

A Blockchain is a type of distributed ledger system that consists of a large number of nodes. A block is a container in which transactions are included, and the transactions can be recorded in chronological order by chaining blocks. To work properly, it is essential that the nodes in the Blockchain system have the same image of the chained-blocks. Blockchain systems use various types of consensus algorithms to achieve the same states among the nodes, and the fundamental elements in these algorithms are proof of work and the main chain selection policy, particularly in permissionless Blockchain systems. However, consensus algorithms for permissioned Blockchain systems can be completely different from those of permissionless blockchain systems. In this paper, we overview the basic working mechanism of consensus algorithms, and briefly introduce a few that are currently being applied.

* DOI: 10.22648/ETRI.2018.J.330105

*본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 정보통신·방송 연구개발 사업의 일환으로 수행하였음[2017-0-00045, 초연결 지능 인프라 원천기술 연구개발].

## Ⅰ. 서론

비트코인(Bitcoin)으로부터 시작된 블록체인 기술은 차세대 인터넷을 이끌 혁신적 기술[1]로서 수많은 업계의 관심을 모으고 있으며, 이더리움(Ethereum), 하이퍼레저(Hyperledger), 리플(Ripple), R3 등의 등장과 함께 빠르게 산업적으로 영향력을 확대해가고 있다. 또한, 가상화폐, 에셋 관리, 공유 경제, IoT, 헬스, 물류 등 다양한 분야에 걸쳐 블록체인 솔루션을 제공하는 신생 기업이 탄생하여 기술 및 플랫폼에 대한 경쟁이 뜨거워지고 있다.

많은 전문가와 기업들은 블록체인 기술을 미래의 인터넷(가치의 인터넷)으로 언급하면서 특히 금융 서비스에서 시장의 판도를 바꿀 혁신적인 기술로 전망하고 있다. WEF(World Economic Forum)에서는 일상생활에 디지털 연결성(Digital connectivity)의 침투로 인해 조만간 사회 전반적으로 큰 변혁을 겪을 것이라고 예상하며, 그 변혁에 선두에 있는 기술 분야 중 하나로서 블록체인 기술을 꼽고 있다[2]. IDC에서는 블록체인 기술로 금융업계의 비용절감 규모가 2022년 약 200억 달러에 달할 것으로 전망했고, 맥킨지에서는 블록체인 기술을 금융시스템에 활용하면 고객 데이터베이스 관리와 보안 등과 관련된 금융비용 절감효과가 연간 23조 원에 이를 것으로 예상했다. Grand View Research의 블록체인 시장 분석 보고서[3]에 따르면 블록체인 관련 시장 규모는 2015년 기준으로 509백만 달러 규모에서 연평균 성장률(CAGR) 37.2%로 성장하여, 2024년에는 7,592백만 달러 규모로까지 빠르게 성장할 것으로 예측되고 있다.

블록체인 시스템은 수많은 노드가 P2P 네트워크로 연결되어 사용자의 트랜잭션을 처리하는 시스템으로서, 트랜잭션에 대한 기록을 순차적으로 저장하는 일종의 분산 장부(DB로 생각하여도 무방하다) 시스템이라고 볼 수 있는데, 한 번 기록된 내용은 변경이 거의 불가능하다는 특징을 가진다. 블록체인 시스템에서는 모든 노드가 동일한 트랜잭션에 대한 처리 기록을 가지도록 하여야 하는데 그것을 가능하게 하는 것이 합의 알고리즘이다.

최초의 블록체인 기술이 적용된 시스템인 비트코인에서는 합의 알고리즘으로서 일명 작업 증명(Proof of Work)과 가장 긴 체인(Longest Chain)을 선택하는 방법이 사용되었다. 비트코인의 합의 알고리즘은 태생적으로 최대 7 TPS밖에 처리할 수 없는 성능의 한계가 있으며, 작업 증명으로 인해 많은 에너지가 낭비된다는 문제를 가지고 있다. 비트코인 이후 등장한 블록체인 시스템은 자신의 시스템에 맞도록 성능 문제 또는 에너지 문제를 완화시키는 변경된 합의 알고리즘을 도입하고 있는 추세이다. 일례로 이더리움의 경우 가장 긴 체인을 메인 체인으로 보는 것이 아니라 가장 많은 서브 트리를 가지는 체인을 메인 체인으로 보는 방법인 GHOST[4]를 응용한 합의 알고리즘을 도입했다.

블록체인 시스템은 크게 블록체인에 참가하는 노드의 참여 제한 여부에 따라 비허가형(Permissionless) 블록체인과 허가형(Permissioned) 블록체인으로 나눌 수가 있다.[1] 비허가형 블록체인의 경우, 대부분 전술한 작업 증명 기반의 합의 알고리즘을 사용하는데, 허가형 블록체인의 경우는 완전히 다른 형태의 합의 알고리즘을 사용할 수 있다. 왜냐하면, 참여하는 노드가 제한되므로 그 점을 활용할 수 있는 알고리즘의 적용이 가능하기 때문이다. 대표적인 알고리즘으로서 PBFT (Practical Byzantine Fault Tolerant)[5]가 있다.

본고에서는 블록체인 시스템에 사용되는 다양한 합의 알고리즘에 대해 살펴보고자 한다.

## Ⅱ. 블록체인 및 합의 알고리즘 개요

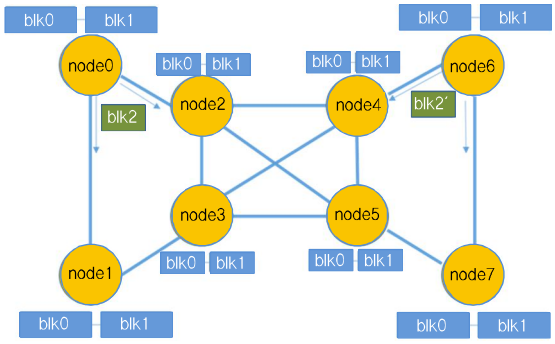
블록체인은 트랜잭션 정보를 기록한 일종의 분산 장

1) 비허가형 블록체인은 퍼블릭 블록체인이라고 불리고, 허가형 블록체인은 운영 방법에 따라 프라이빗 블록체인 또는 컨소시움 블록체인으로 나뉠 수 있다.

부로서 각 노드가 각각 자신의 장부를 가지고 있고, 각 장부의 내용은 합의 알고리즘에 의해 동일하게 유지된다. 장부에 기록되는 하나의 엔트리는 트랜잭션으로 표현될 수 있으며, 장부에 기록을 원하는 사용자가 트랜잭션을 생성하여 P2P 네트워크에 전달하면 블록체인 처리 노드들이 이를 모아 블록을 생성한다. 블록이 서로 체인으로 연결되어 있기 때문에, 트랜잭션의 순서화된 기록이 가능하게 된다. 이렇게 연결된 블록체인의 인스턴스는 하나의 분산 장부를 표현하게 된다.

분산 장부의 일관성 혹은 동일성은 결국 각 노드가 가지고 있는 블록체인 이미지의 동일성에서 비롯하게 된다고 할 수 있다. 블록체인 이미지의 동일성은 만약 중앙의 한 노드가 블록을 전담하여 생성한다고 하면, 자연스럽게 유지될 수 있을 것이다. 그러나, 블록체인 기술의 핵심은 특정 노드를 신뢰하지 않으면서 신뢰를 제공한다는 것이기 때문에, 중앙집중적 방식이 아닌 개별 노드들이 자율적으로 블록을 생성하되 일종의 합의 과정을 거쳐 결국에는 모든 노드가 같은 블록체인 이미지를 가지도록 하는 방식을 사용한다.

(그림 1)은 전형적인 블록체인 시스템의 블록 생성 과정을 도시한다. 각 노드가 블록0과 블록1로 구성된 동일한 블록체인을 가지고 있다고 하자. 이때 각 노드는 블록1에 연결할 유효한 새로운 블록(블록2)을 만들기 위해 작업 증명을 진행하게 된다. 노드0과 노드6이 비슷한 시기에 블록2와 블록2′를 생성하는 데 성공하였다

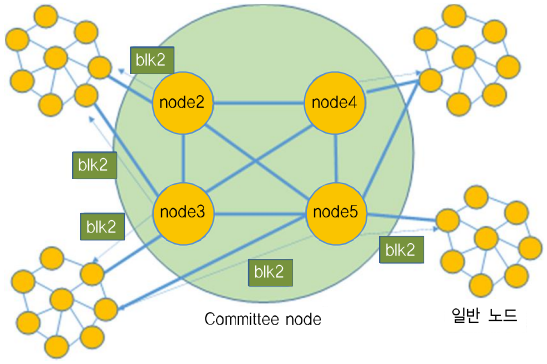



(그림 1) 블록 생성 과정

면, 노드 간 네트워크 토폴로지와 전송속도에 따라 특정 시점에서 노드에 따라 블록체인이 다르게 구성될 수 있다.

예를 들어 노드0, 노드1, 노드2, 노드3는 블록2를 블록1의 자식으로 연결하고, 노드6, 노드7은 블록2′를 블록1의 자식으로 연결할 수 있으며, 노드4와 5는 블록2와 블록2′ 모두를 블록1의 자식으로 연결하고 있을 수 있다. 이렇듯 한 부모에 대해 자식 블록이 여러 개가 생기는 상황을 일명 'fork'라고 하는데, 'fork'가 된 상황은 노드 간 블록체인으로 구성된 분산장부가 일치하지 않는 상태를 유발하게 된다.

따라서 이러한 불일치를 해결하기 위해서는 모든 노드가 블록2가 블록1의 자식이 될지 블록2′가 자식이 될지 동일하게 결정할 수 있어야 하는데, 이것을 해결하는 대표적인 방법이 비트코인에서 쓰이는 가장 긴 체인을 선택하는 방법이다. (그림 1)에서 노드2가 블록2를 부모로 하여 블록3을 만들고 이것이 모든 노드에게 전파가 된다고 하면, 블록2는 자식 블록이 연결됨으로서 블록2′보다 긴 체인을 형성하게 되어 블록2가 속한 체인이 메인 체인이 된다. 이때, 각 노드는 특별한 자격제한 없이 블록을 만들 수 있으므로 각 노드에서 블록을 아무런 노력 없이 쉽게 만들 수 있게 된다면, 동시에 수많은 블록이 만들어질 수 있으며, 이는 각 노드가 하나의 동의된 블록체인에 합의하는 것을 거의 불가능하게 만들 수 있다. 따라서 각 노드 중에서 일종의 리더 노드를 선택하여 그 노드로 하여금 블록을 만들도록 하여야 하며, 그 리더 노드를 결정짓는 방법이 바로 작업 증명이라고 할 수 있다. 비슷한 시점에 다수의 리더 노드가 생길 경우, 'fork'가 생길 수 있는데 이것은 전술한 방법으로 해결한다. 정리하면 전형적인 블록체인 시스템의 합의 알고리즘은 랜덤하게 하나 또는 소수의 블록을 생성할 리더를 선택하는 작업 증명과 리더 간 다른 블록을 생성하였을 경우 이를 해결하는 정책(예: Longest chain)으로 이루어진다고 할 수 있다. 퍼블릭 블록체인에서는 다수의 노드가 블록을 생성하기 위해 서로 경쟁하게 되는데, 이는 블록




(그림 2) 허가형 블록체인 시스템 개념 구조

생성에 성공하면 인센티브를 받을 수 있기 때문이다. 이러한 특성 때문에, 블록을 만드는 것을 마이닝(Mining)한다고 하며, 블록을 마이닝하는 노드를 마이너(Miner)라고 일컫는다.

작업증명은 비허가형 블록체인에서 필수 불가결한 것이지만, 성능 문제와 에너지 낭비 문제를 가지고 있다. 최근에는 이런 문제를 극복하기 위해 지분 증명(Proof of Stake)이나 경과 시간 증명(Proof of Elapsed Time) 등의 알고리즘이 개발되고 있으며, 샤딩(Sharding) 기법을 적용하는 연구 또한 진행되고 있다.

허가형 블록체인의 경우, 블록체인 노드를 일정 부분 신뢰할 수 있다는 측면에 의해 비허가형 블록체인 시스템에서 사용되는 합의 알고리즘과는 전혀 성격이 다른 알고리즘을 적용할 수 있다. (그림 2)는 허가형 블록체인의 개념적인 구조를 도시한다. 즉 블록을 생성할 수 있는 자격을 가진 노드를 미리 정하여 놓고, 그 노드들로 구성된 위원회(Committee)를 구성하여 그 위원회에서 멤버들 간 합의를 통해 하나의 합의된 블록을 생성하고, 이를 다른 노드들에게 전파하는 방식을 사용하는 것이 가능하다.

## Ⅲ. 합의 알고리즘

### 1. 작업 증명(PoW: Proof of Work)

작업 증명은 Satoshi Nakamoto의 논문 'Bitcoin: A Peer-to-Peer Electronic System[6]'에 처음 소개된 메커니즘으로, 블록 생성을 하고자 하는 노드들이 특정한 해시(hash) 값을 찾는 연산을 수행하여 특정한 난이도의 작업을 수행했음을 증명하는 것이다. 마이너들은 해시 값을 찾기 위해 경쟁을 하고, 특정 마이너가 목표 값에 해당하는 해시 값을 찾는 데 성공하면 블록이 생성된다.

비트코인에서 블록($B$)의 해시 값은 $\text{hash}(B) \leqslant M/D$로 정의되는데[7], 여기서 $D$는 난이도(difficulty)이고 $M$은 난이도 $D$의 최대 값($2^{256}-1$)으로, 마이너들은 반복적으로 조건을 만족하는 블록 $B$의 해시 값을 찾게 된다. 작업 증명에서 높은 컴퓨팅 파워를 가질수록 빠른 속도로 해시 값 계산을 할 수 있는데, 비트코인의 경우 평균 10분이 소요되도록 설계되었다. 비트코인에서는 기술의 발전이나 많은 컴퓨팅 자원을 이용하는 방식으로 컴퓨팅 파워를 올려 해시 암호를 푸는 시간을 단축할 수 문제에 대응하기 위해, 해시 난이도를 주기적으로 조절하여 블록 생성 주기를 일정하게 유지하도록 하고 있다.

비트코인에서는 블록 생성이 개별 노드에서 자율적으로 수행되기 때문에, 작업 증명을 통해 랜덤하게 블록을 생성할 노드를 선택하게 되더라도, 같은 부모를 가지는 두 개 이상의 자식 블록이 거의 동시에 생성되는 fork가 발생할 수 있다. Fork 현상은 노드 간 분산 장부 내용에 불합의가 있다는 의미로 비트코인에서는 가장 긴 체인이 최종적으로 승자가 되도록 함으로써, 분산 장부의 불일치 문제를 해결하고 있다.

이더리움 홈스테드(Homestead) 버전은 비트코인보다 빠른 블록 생성 주기를 가지는데, 이로 인해 블록이 동시에 생성될 확률이 높아지게 되어 fork가 발생할 확률 또한 높아지게 된다. 잦은 fork의 발생은 결국 동일한 블록체인을 가지는 합의에 이르는 것을 어렵게 만들며, 합의에 이르렀다 하더라도, 가장 긴 체인에 포함되지 못한 많은 블록이 버려지게 되어 시스템 전체적으로 효율을 떨어뜨리는 문제가 발생될 수 있다. 이렇게 메인 체

인에 붙지 못하고 버려지는 블록을 비트코인에서는 고아 블록이라고 하는데, 이더리움에서는 고아 블록을 엉클 블록으로 칭하고, 메인 체인에 이들을 포함시키는 GHOST[4] 계열의 알고리즘 사용하여 전술한 문제를 해결하였다.

이더리움 GHOST의 핵심은 메인 체인이 가장 긴 체인이 아니라 가장 무거운(heaviest) 체인이 된다는 것이다. 가장 무거운 체인이란 메인 체인을 결정할 때 자식 블록뿐만 아니라, 엉클 블록을 카운트하여 메인 체인의 무게에 포함시켜 계산함으로서, 가장 무거운 체인을 선택하는 메커니즘으로 마이닝 후에 버려지는 블록을 줄이는 효과를 가진다. 여기서, 엉클 블록에 대한 개수 제한이 없으면 특정 블록에 대해 유효한 엉클 블록 계산이 복잡해지므로, 이더리움은 최대 7세대까지만 엉클 블록을 포함하도록 한정한다.

### 2. 지분 증명(PoS: Proof of Stake)

지분 증명 방식은 작업 증명 방식의 과도한 에너지 소비 문제 해결을 위한 대안으로 제시되었으며, 참여자의 소유 지분이 블록 생성권 지분에 반영이 되는 합의 알고리즘이다. 작업 증명에서는 마이너의 컴퓨팅 파워에 따라 블록 생성 확률이 높아지나, PoS에서는 마이너가 보유하고 있는 화폐의 양에 비례하여 블록을 생성하게 된다.

PoS에서 해시 함수는 아래와 같이 정의된다[7].

$$\mathrm{hash}(\mathrm{hash}(B_{\mathrm{prev}}),A,t)\leqslant \mathrm{bal}(A)M/D.$$

$B_{\mathrm{prev}}$는 이전 블록, $A$는 계정(address), $t$는 타임스탬프, bal($A$)는 address $A$가 현재 소유한 balance, $D$는 난이도, $M$은 $D$의 최대값을 의미한다. 블록 $B$의 해시 값은 $A$가 소유한 balance와 난이도의 영향을 받는다. 따라서 많은 지분 소유자가 쉬운 난이도의 문제를 풀게 된다.

PoS 개념은 2011년 Bitcointalk 포럼에서 처음 제안되었으며, PoS 기반 가상화폐에는 Peercoin, Nxt, Novacoin 등이 있다. PoS 방식은 초기에 많은 지분을 보유한 자가 블록을 생성할 확률이 높으므로, 시간이 지날수록 초기에 지분을 많이 가진 자에게 유리해지는 불평등 문제가 발생하였다. 이를 해결하기 위해서, 보유한 코인의 양과 코인 보유 일수(Coin age)를 기반으로 참여자에게 블록 생성권을 주고, 블록 생성에 대한 보상으로 코인을 주는 방식이 고안되었다. Peercoin이 이러한 방법을 적용한 대표적 예이다. 예를 들어, Bob이 Alice에게 10코인을 받고 90일간 보유했다면, Bob은 900 coin-day를 누적하였고, Bob이 Alice에게 받아 90일간 보유한 10코인을 사용하면, Bob이 누적한 900 coin-day를 소비한 것이다. Peercoin은 코인 보유 일수를 기반으로 하기 때문에, 보유 코인이 적어도 코인 보유 일수가 길면 블록 생성 가능성이 커지므로, 보다 균등하게 블록 생성 기회를 줄 수 있는 장점이 있다[8].

코인 보유 일수 개념을 적용하더라도, 적은 양의 코인이라도 장기간 보유하여 코인 보유 일수를 늘린 다음, 연속적으로 블록을 생성하는 형태의 공격을 할 수 있다. 이러한 문제를 극복하기 위해 Novacoin에서는 코인 보유 일수에 가중치(Weight)를 두는 방법을 도입하였다. 가중치를 주는 방식은 예를 들어, 코인 보유 일수가 1~30인 경우는 가중치 0, 코인 보유 일수가 30~60인 경우는 가중치 30, 코인 보유 일수가 60~90인 경우는 가중치 60, 코인 보유 일수가 120일 이상인 경우는 가중치 120으로 할당한다.

PoS 방식은 블록 생성 주기를 단축시킬 수 있고 컴퓨팅 파워 낭비를 줄일 수 있어 리소스 관점에서는 효율적이나, 초기 코인 분배 문제(Initial Distribution Problem)와 Nothing at Stake 문제가 발생할 수 있다. 초기 코인 분배 문제는 PoS 방식에서 블록 생성 지분이 소유 지분을 기반으로 하기 때문에 초기에 코인을 많이 보유한 참여자가 블록 생성에 유리하다는 점에 의한 공정성에 대한 문제이며, Nothing at Stake는 유효한 블록체인이 두 개 이상 존재하는 fork 상황에서 참여자들이 보상받

을 확률을 높이기 위해 두 개 이상의 블록체인 상에서 블록을 생성함으로써, 하나의 블록체인으로 수렴해 가는 것을 어렵게 하는 것을 말한다. 이러한 상황에 공격자가 뇌물을 주고 유효한 블록체인을 임의로 바꿀 수 있으며, 유효한 블록체인에 대한 합의를 빨리 이루지 못하는 문제가 있다. 따라서 최근에는 이를 보완한 위임된 지분증명(DPoS: Delegated proof of stake) 방식이 제안되어 이용되고 있다.

지분 증명 방식은 일정한 지분을 가진 모든 노드에게 블록 생성 권한을 주었던 반면, DPoS에서 지분 보유자들은 지분에 비례한 투표로 대표자를 선출하고, 대표자들에게 블록 생성과 검증에 대한 권한을 부여하여 합의에 대한 권리를 위임한다. 투표에 의해 선출된 대표자들이 블록을 생성하기 때문에 합의에 걸리는 시간과 비용이 적게 소요되고, 작업 증명과 지분 증명에 비해 상대적으로 단위 시간 동안 생성되는 블록의 개수도 많다. DPoS 방식은 Tendermint, Slasher, BitShares, Ethereum Casper 등에서 사용하고 있다[7].

Tendermint는 블록체인 어플리케이션 개발 및 블록체인 간 통신을 위한 코스모스(Cosmos) 플랫폼을 제공하며, PBFT 계열 합의 알고리즘을 사용하여 대표자인 검증인들(validators)의 투표를 통해 합의가 이루어진다. Tendermint는 블록체인의 각 높이에서 블록을 결정하는데, 라운드를 기반으로 제안자의 블록에 대해서 2단계의 투표(Prevote와 Precommit)를 하여, 2/3 이상의 합의가 있으면 다음 단계로 진행하게 된다. Tendermint에서는 잠금(lock) 메커니즘을 사용하는 데, 검증인이 특정 블록에 투표를 하면 투표권을 잠가서 동일 높이의 다른 블록에 투표를 못하게 함으로써 악의적인 행동을 막을 수 있다[9].

BitShares는 엔터프라이즈용 스마트 컨트랙트 플랫폼인 Shares 2.0을 제공하며, 컬러드 코인과 같이 다양한 디지털 자산을 생성하여 이용할 수 있다. BitShares에서 블록 생성은 증인(witness)이 수행하고, 대표자(delegate)는 네트워크 정보 변경 권한이 있다. 지분 보유자들은 소유한 지분에 비례하여 증인 선정 투표를 하고, 투표 결과 상위 $N$명의 증인을 선출한다. 선출된 증인들은 순번을 정하여 차례대로 매 2초마다 블록을 생성하고, $N$명의 순서가 끝나면 증인 목록이 섞여 블록 생성 순서를 바꾸게 된다. BitShares에서 지분 보유자들로부터 선출된 대표자들은 증인에 대한 보상, 트랜잭션 비용, 블록 크기, 블록 주기를 포함하는 네트워크 파라미터를 변경할 권한을 가진다. 네트워크 정보가 변경되면, 대표자들은 특별한 계정(genesis account)에 공동 서명하여 승인 절차를 거친다. 증인들은 블록 생성에 대한 보상을 받지만, 대표자들은 노력에 대한 보상이 없는 차이가 있다.

이더리움 캐스퍼(Casper)는 사전에 선택된 검증인들이 블록 생성 및 합의 결정 권한을 가진다. 블록 생성은 사전에 지정된 검증인들이 트랜잭션을 수집하여 블록을 생성한 후 서명하여 네트워크로 보내게 된다. 블록 합의 결정을 위해서 검증인들은 특정 블록에 대해서 보증금을 예치하고 베팅을 하게 된다. 특정 높이에서 하나의 블록을 결정하기 위해서 2/3 이상의 검증인들이 베팅을 하여 합의를 이룰 때까지 베팅 라운드를 수행하게 된다(Consensus by bet). 검증인들은 자신이 베팅한 블록이 최종 블록으로 선택되면 보상을 받게 된다. 이러한 베팅은 캐스퍼 컨트랙트로 구현이 되며 베팅 생성, 베팅 라운드 참가 및 철회 기능을 제공한다[10].

최근의 PoS 방식은 Nothing at Stake 문제를 해결할 수 있는 DPoS 형태로 개발되고 있는 추세이며 실제 환경에서 DPoS 방식에 대한 검증이 뒷받침되어야 한다.

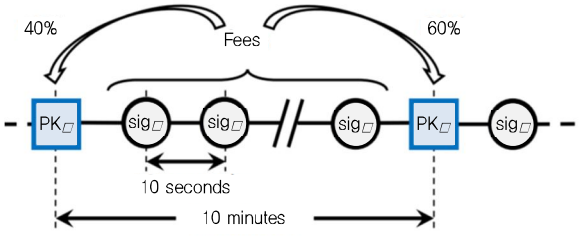
### 3. Bitcoin-NG

Bitcoin-NG[11]는 비트코인의 메인 개념이라 할 수 있는 작업 증명과 가장 긴 체인 선택은 유지하면서도 약간의 프로토콜 구조 변경을 통해 TPS 관점에서의 성능 향상을 도모한 프로토콜이다. Bitcoin-NG는 'key block'

과 'micro block'이라는 두 가지의 블록 유형을 사용한다. key block 생성은 일종의 블록을 만들 자격을 얻는 과정이고 micro block은 자격을 획득한 노드(즉, key block을 생성한 노드)가 트랜잭션을 처리하기 위해 생성하는 블록이다. 기존 비트코인에 비해 key block을 생성한 노드는 다수의 micro block을 작업 증명 과정 없이 보다 빠른 주기로 생성할 수 있는데, 이로 인해 TPS의 향상을 가져올 수 있다.

Key block을 생성한 노드를 일종의 리더 노드라고 할 수 있는데, 일련의 micro block이 합법적인 리더에 의해 생성되었음을 보증하기 위해 micro block에는 리더 노드의 디지털 서명이 포함된다. 리더 노드는 다른 노드가 key block 마이닝에 성공하게 되면 해당 노드에 의해 대체된다. (그림 3)은 이러한 과정을 보여준다.

Bitcoin-NG도 비트코인과 유사하게 fork가 발생할 수 있다. Key block 마이닝은 여전히 비트코인과 동일한 방식으로 작업 증명을 통해 이루어지므로, 동시에 여러 개의 key block이 마이닝 될 수 있기 때문이다. Bitcoin-NG에서는 새로운 key block을 생성한 노드가 보다 많은 트랜잭션을 처리하여 그 수수료를 챙김으로써 이득을 얻기 위해 이전 key block을 생성한 노드가 생성한 micro block을 일부러 배제하는 것이 가능하다. 이를 막기 위해 Bitcoin-NG에서는 새로운 key block을 마이닝한 노드에 이전 micro block의 트랜잭션 수수료 중 일부를 나누어 주는 전략을 취한다. Bitcoin-NG의 또 하나의 단점은 리더 노드가 악의적이라면 이중 지불 공격을 하는 것이 가능하다라는 점이다.

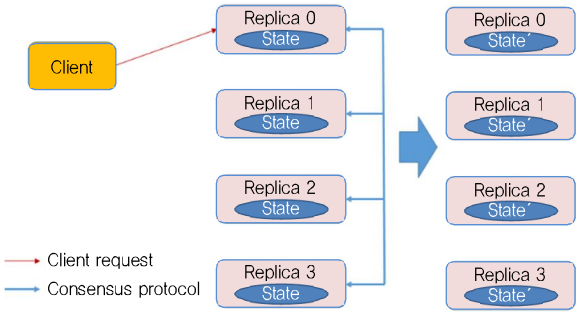



(그림 3) Bitcoin-NG의 블록 생성 과정[11]

## 4. BFT 계열

블록체인 시스템은 트랜잭션 정보를 기록한 일종의 분산 장부 시스템으로 볼 수 있으며, 블록체인에 참여하는 모든 노드에 의해 동일한 분산 장부 복사본이 관리된다. 이러한 측면에서 블록체인 시스템은 전통적인 state machine replication 시스템과 매우 유사한 특징을 가진다고 볼 수 있다.

(그림 4)는 State Machine Replication 시스템을 개념적으로 도시한다. 시스템은 여러 개의 리플리카(replica)들로 구성되고, 각각의 리플리카는 각각 자신만의 상태(state)를 가지고 있다. 각각의 리플리카는 결정적인(deterministic) 서비스 로직을 수행하는 state machine으로 볼 수 있으며, 현재 state에서 특정 명령이 성공적으로 수행되면 다른 state로 전이한다. 클라이언트가 리플리카에 특정 연산(예: 특정 변수에 값을 쓰라고 요청)을 요청하면 리플리카들이 합의 프로토콜을 이용하여 같은 연산을 수행하고 최종적으로 모든 리플리카들이 동일한 state를 가지기 위해 노력한다. 블록체인의 경우 각각의 계정 정보를 state를 구성하는 하나의 변수로 볼 수 있으며, 특정 순간에서의 모든 계정 정보의 현재 값이 리플리카 시스템의 현재 state를 결정한다고 볼 수 있다. 블록이 하나 늘어나는 경우 리플리카 시스템의 state가 state'로 바뀐다고 생각할 수 있다. 이러한 유사점 때문에 블록체인에 리플리카 시스템에서 사용하는 합의 프로토콜을 적용하는 것이 가능하고, 이미 여러 블




(그림 4) State machine replication

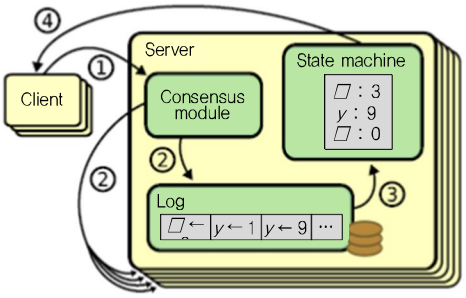
록체인 시스템에 적용되고 있다.

이러한 리플리카 시스템은 싱글 호스트 시스템에게 고가용성을 제공할 수 있지만, 리플리카 간의 네트워크 오류, 리플리카 자체의 고장 등의 문제로 여전히 정상 동작되지 않을 수 있는데, 합의 프로토콜은 이러한 문제들을 다룰 수 있어야 한다. 합의 프로토콜은 일반적으로 다음 두 개의 시스템 속성을 만족시켜야 한다.

- Safety: 시스템에 나쁜 일이 발생하지 않는다는 의미이며, 모든 정상적인 리플리카는 같은 상태에 동의하여야 하고, 그 상태는 유효해야 함.
- Liveness: 시스템은 항상 살아 있어야 한다는 의미이며, 결국에는 어떤 상태에 동의하여야 하고, 모든 리플리카는 동의된 상태에 도달해야 함.

리플리카의 비정상적인 상황을 크게 두 가지로 구분할 수 있는데, 하나는 fail-stop이고, 다른 하나는 비잔틴 폴트(Byzantine fault)이다. Fail-stop은 단순히 노드가 고장이 나서 멈추는 형태의 오동작인 반면 비잔틴 폴트는 리플리카가 악의적인 행동을 포함하여 임의의 동작을 할 수 있는 것을 의미한다. Fail-stop 형태의 고장을 가정한 대표적인 합의 프로토콜에는 Paxos[12], [13]와 Raft[14]가 있고, 비잔틴 폴트를 가정한 대표적인 합의 프로토콜로는 PBFT[5]가 있다. Paxos는 합의 알고리즘 중 초기에 만들어진 것으로, 여러 곳에서 사용되고 있다. 예를 들어, 구글의 Chubby lock 서비스[15]와 Megastore 및 Spanner[16]에 스토리지 시스템에 사용되고 있으며, Microsoft의 Autopilot 서비스[17]와 윈도우 Azure 스토리지[18]에 사용되고 있다.

Paxos는 고장 감내 분산 시스템에서 여러 프로세스 간에 하나의 값에 동의하기 위한 프로토콜로서 동시에 여러 개의 값이 제안되지만, 결국에는 이 중 하나의 값이 선택되도록 하는 것이 골자이다. 이를 위해 각각의 프로세스는 값을 제안하는 제안자(proposer), 제안된 값

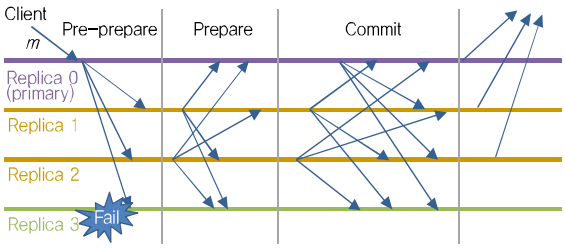



(그림 5) Raft reference architecture[14]

에 투표하는 수용자(accepter), 제안된 값을 배워가는 학습자(learner)의 역할을 수행할 수 있다. 여러 제안자에 의해 동시에 여러 값의 후보가 제안될 수 있다는 점 때문에 프로토콜의 동작이 복잡하다.

Raft는 비교적 최근에 개발된 알고리즘으로서, Paxos를 보다 이해하기 쉽게 만들기 위해 고안되었다. Raft는 기본적으로 (그림 5)와 같은 복제된 state machine 구조를 가진다. 클라이언트의 요청에 대한 처리 결과는 합의 모듈에 의해 결정되어 로그에 쓰여지고, 개별 리플리카의 state machine을 로그의 값에 따라 상태를 변경하는 모델이다. Raft에서는 클라이언트 요청을 하나의 리더 노드가 처리하여 로그를 업데이트하고, 이 로그가 다른 리플리카에도 반영되도록 하는 형태로 동작한다. 리더가 문제가 있을 경우, 리더는 리더 선출 프로토콜에 따라 새롭게 선출된다. Viewstamped Replication[19], Zab(ZooKeeper Atomic Broadcast)[20] 프로토콜도 같은 계열의 프로토콜로 볼 수 있다.

PBFT 프로토콜은 1982년에 발표된 논문인 비잔틴 장군 문제[21]를 해결하기 위한 실질적인 프로토콜로서 제안되었다. 비잔틴 장군 문제란 도시를 포위하고 있는 여러 장군이 하나의 작전(공격 또는 후퇴)에 어떻게 동의할 수 있는가를 다루는 문제로서, 장군이 악의적인 행동을 할 수 있다는 점을 가정하고 있다. 이러한 이유로 BFT 계열의 프로토콜이라 함은 단순 고장난 노드 뿐만 아니라 악의적인 노드가 있음에도 불구하고 전체 시스템이 안정적으로 동작하도록 하는 프로토콜을 일컫는




(그림 6) PBFT 프로토콜 메시지 흐름도

다. PBFT는 BFT계열 프로토콜 중 실용적으로 쓰일 수 있는 가장 대표적인 프로토콜이다.

(그림 6)은 PBFT 프로토콜의 동작을 간단하게 보여준다. 리플리카 중 의사결정의 리더 역할을 하는 primary 노드가 있으며, primary 노드의 주도하에 순차적으로 명령이 수행될 수 있도록 한다. 만약 primary 노드가 고장이 나거나 악의적인 행동을 하게 되면 일명 'view change'라는 절차를 통해 primary 노드를 바꾼다. 일반적으로 여러 개의 리플리카로 구성되는 리플리카 시스템에서 전체 노드($N$개) 중 몇 개의 노드가 문제가 있을 시($F$개) 정상 동작하도록 설계가 되는데, PBFT 프로토콜의 경우, $N$=3$F$+1일 때 정상 동작이 보장되는 프로토콜이다.

최근에는 전술한 프로토콜들이 특히 허가형 블록체인 시스템에 적용되고 있다. 하이퍼레저 패브릭(Fabric)의 경우, 0.6 버전에서는 PBFT 프로토콜을 합의 알고리즘으로 사용하였으며, 1.0의 경우에는 Kafka[22] 기반의 순서제공 서비스(Ordering service)에 기반한 합의 알고리즘을 사용하고 있다. Kafka는 Zab 프로토콜을 사용하는 ZooKeeper[23] 기반으로 동작한다. Tendermint[24]는 변형된 PBFT 프로토콜을 사용하고 있고, R3 Corda[25]의 경우 트랜잭션 처리시 서기 서비스(Notary service)에 의존하는데, 서기 서비스는 Raft나 PBFT 프로토콜에 의해 고장 감내적으로 동작하도록 할 수 있다.

정리하면, Paxos나 BFT 계열의 프로토콜은 현재 블록체인 시스템에서 합의 알고리즘에 직접적으로 도입되어 쓰이거나 합의 알고리즘에 있어서 중추적 역할을 하는 노드의 고장 감내를 보장하기 위해 쓰이고 있다.

### 5. 경과 시간 증명(PoET: Proof of Elapsed Time)

작업 증명 방식의 경쟁적 해싱 연산으로 낭비되는 에너지를 줄이면서 작업 증명과 유사한 Security를 보장하기 위해 최근에 제시된 방식으로 '경과 시간 증명(PoET: Proof of Elapsed Time)'이 있다. PoET는 하이퍼레저 쏘투스 레이크(Sawtooth Lake)에서 제안된 합의 알고리즘으로 신뢰할 수 있는 보안 모듈(Intel SGX)을 기반으로 블록을 생성하는 리더를 랜덤하게 선정하는 방식이다. PoET는 분산 합의를 효율적으로 이루기 위해 가능한 다수의 노드가 합의에 참여하여 공정하게 리더를 선정하도록 하며, 하드웨어의 성능에 의존하는 작업 증명 방식과 달리 보안 CPU 명령어를 사용하여 리더를 선정함으로써 안전성과 무작위성(Randomness)을 보장한다[26].

PoET의 경우 마이너의 동작을 시스템 보안 모듈인 enclave에서 수행되도록 함으로써, 원천적으로 악의적인 노드의 개입을 막으며, 블록을 생성할 리더를 선출하는 방법은 단순히 리더 선출 코드 호출 시에 enclave에서 랜덤한 시간만큼 기다리도록 하고, 가장 먼저 반환된 노드를 리더로서 선출하는 방식이다. 리더 선출이 올바른지에 대한 유효성 검사는 SGX를 이용하여 할 수 있다. PoET는 리더 선정 참가 비용이 낮아 다수의 검증인들이 참여할 수 있으므로 합의 알고리즘의 견고성이 증가하는 장점이 있으나, Intel SGX에 의존하는 단점이 있다.

### 6. Sharding 기법

최근 블록체인의 성능을 향상하기 위한 목적으로 트랜잭션의 병렬 처리가 가능한 샤딩(Sharding) 기법을 도입하려는 시도가 일어나고 있다. 샤딩은 일반적으로 데이터베이스에서 효율적으로 확장성을 확보하기 위해

사용하는 기법으로, 전체 DB를 조각내어 각 조각이 다수의 각기 다른 사이트에 의해 처리되도록 하는 기법이다. 블록체인에 샤딩 기법을 적용함에 있어서의 문제는 어떤 노드가 어떤 샤드를 담당하게 할 것인가의 문제와 각 샤드에서 어떤 트랜잭션을 담당하게 할 것인가의 문제로 볼 수 있다. 예를 들어 샤드와 노드의 매핑 문제는 정적인 방법을 사용하는 것도 가능할 것이다. 즉 노드가 사전에 어떤 샤드를 담당할 것인지를 지정하는 방법이다. 그러나 정적으로 샤드를 지정하는 것은 퍼블릭 블록체인의 개방성에 위해가 될 수 있으며 보안 측면에서도 문제가 있기 때문에, 동적으로 샤드와 노드를 매핑하는 방법이 연구되고 있다. 트랜잭션과 샤드의 매핑은 블록체인에 기록되는 변수(예: 계좌)를 이용하는 방법이 사용될 수 있다. 예를 들어 A라는 계좌와 관련된 트랜잭션은 샤드 A에서 처리하고 B라는 계좌와 관련된 트랜잭션은 샤드 B에서 처리하는 것이다. 샤드는 트랜잭션을 병렬 처리할 수 있다는 점에서 성능향상을 기대하고 있지만, 여러 샤드에 걸쳐서 처리해야 하는 크로스 샤드 트랜잭션이 존재하고, 상황에 따라 각 샤드에서 처리한 트랜잭션에 대해 전역적인 관점에서 conflict가 없는지를 검사해야 하기 때문에 일반적으로 성능이 샤드 갯수에 따라 선형적으로 증가하지 않는다.

샤딩 적용 연구 중의 하나인 Elastico[27]는 마이닝노드들을 로컬 위원회(Committee)라고 불리는 소규모 그룹들로 분할하여 각 로컬 위원회가 서로 다른 트랜잭션 집합을 병렬 처리하도록 하는 방법을 제시한다. 로컬 위원회의 구성은 'Sibyl attack'을 방지하기 위해 작업 증명을 통해 랜덤하게 결정한다. 이렇게 결정된 로컬 위원회는 트랜잭션을 검증하는 검증 노드 역할을 수행하게 되는데, 이 때 검증 작업은 PBFT 프로토콜을 이용하여 로컬 위원회 멤버들 간 합의를 통해 이루어진다. 이렇게 병렬적으로 처리되어 검증된 트랜잭션들은 최종적으로 글로벌 위원회(Final Committee)에 의해 다시 병합된 후에 블록으로 생성되어 브로드캐스팅 됨으로써 분산된 블록체인에 업데이트된다. Elastico에서의 이러한 샤딩 프로토콜 방식은 단지 트랜잭션 처리에 대한 부하만 병렬로 분배할 수 있는 구조이기 때문에 모든 블록 검증 노드들이 여전히 중복하여, 모든 정보가 포함된 온전한 블록체인 기록을 유지해야 한다.

이더리움은 확장성 및 느린 지연시간을 개선하고자 블록체인 상태를 샤딩하기 위한 여러 전략을 연구 중이다. 이더리움의 블록체인 샤딩기법은 무작위로 선택된 네트워크 노드 집합을 구성하여 특정 계정의 접두사와 관련된 트랜잭션만을 검증하는 구조로, 여러 조각으로 분리된 블록체인 상태가 네트워크의 다른 노드들에 분산되어 저장된다. 그리고 한 샤드의 트랜잭션이 다른 샤드에 영향을 줄 수 있는 크로스 샤드 통신을 위한 메시지 전달 메커니즘으로 영수증(receipt) 패러다임 접근법을 적용하여, 샤드 내의 트랜잭션이 실행될 때 분산된 형태로 저장되는 '영수증'을 생성하면서 자신의 로컬 샤드의 상태를 변경한다[28].

전반적으로, 블록체인의 샤딩 연구에서 하위 네트워크의 동적구성과 함께 블록체인 샤딩에 대한 크로스 샤드 통신은 매우 어려운 문제이다. 또한, 여러 조각으로 분리된 블록체인 상태가 다른 노드들에 분산 저장되어 관리되는 샤딩 기법에서는, 공격 노드가 단일 샤드만 변조시켜도 전체 블록체인 원장에 큰 영향을 미치는 문제가 발생하므로 이에 대한 이슈를 해결할 수 있는 연구도 필요하다.

## Ⅳ. 결론

블록체인 시스템을 특징짓는 것 중 하나는 합의 알고리즘이다. 비트코인에서 쓰이는 합의 알고리즘은 수많은 분산된 노드들이 하나의 블록체인 이미지를 가질 수 있도록 하는 마법과 같은 역할을 수행해 냈지만, 성능과 에너지 낭비 측면에서 많은 단점을 가지고 있다. 따라서

비트코인 이후의 시스템들은 성능과 에너지 문제를 해결하기 위해 다양한 다른 특성을 가진 독특한 방식의 합의 알고리즘을 도입하고 있다. 합의 알고리즘은 성능, 에너지, 보안, 개방성이라는 가치 중 어디에 중점을 두는가에 따라 다양한 형태로 개발될 수 있으며, 새로운 합의 알고리즘을 개발하기 위한 노력은 앞으로도 계속될 것으로 예측된다.

## 약어 정리

| | |
|---|---|
| BFT | Byzantine Fault Tolerance |
| DPoS | Delegated proof of stake |
| GHOST | Greedy Heaviest Observed Subtree |
| GVR | Grand View Research |
| IoT | Internet of Things |
| P2P | Peer to Peer |
| PBFT | Practical Byzantine Fault Tolerant |
| PoS | Proof of Stake |
| PoW | Proof of Work |
| TPS | Transaction Per Second |
| WEF | World Economic Forum |
| Zab | ZooKeeper Atomic Broadcast protocol |

## 참고문헌


[1] 돈 탭스콧, 알렉스 탭스콧, "블록 체인 혁명," 서울: 을유문화사, 2017.

[2] World Economic Forum(WEF), "Deep Shift: Technology Tipping Points and Societal Impact," SurveyReport, 2015. 9.

[3] Grand View Research, "Blockchain Technology Market," 2017.

[4] Y. Sompolinsky and A. Zohar, "Secure High-Rate Transaction Processing in Bitcoin," In *International Conference on Financial Cryptography and Data Security*, Berlin, Heidelberg: Springer, 2015.

[5] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," *Proc. Symp. Oper. Syst. Des. Implementation*,, New Orleans, LA, USA, Feb. 1999, pp. 1-14.

[6] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," Oct. 2008, Accessed 2017. http://nakamotoinstitute.org/static/docs/bitcoin.pdf

[7] BitFury Group, "Proof of Stake Versus Proof of Work White Paper," 2015. 9.

[8] S. King and S. Nadal, "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake," Self-Published Paper, 2012. 8.

[9] Tendermint Wiki, "Byzantine Consensus Algorithm," Accessed 2017. https://github.com/tendermint/tendermint/wiki/Byzantine-Consensus-Algorithm

[10] Tendermint Wiki, "Ethereum Casper Version 1 Implementation Guide," Accessed 2017. https://github.com/ethereum/research/wiki/Casper-Version-1-Implementation-Guide

[11] I. Eyal, A.E. Gencer, E.G. Sirer, and R. van Renesse, "Bitcoin-NG: A Scalable Blockchain Protocol," *Proc. Usenix Conf. Netw. Syst. Des. Implementation*, Santa Clara, CA, USA, Mar. 2016, pp. 45-59.

[12] L. Lamport, "The Part-Time Parliament," *ACM Trans. Comput. Syst.*, vol. 16, no. 2, May 1998, pp. 133-169.

[13] L. Lamport, "Paxos Made Simple," *ACM SIGACT News*, vol. 32, no. 4, Dec. 2001, pp. 18-25.

[14] D. Ongaro and J.K. Ousterhout, "In Search of an Understandable Consensus Algorithm," *USENIX Annu. Technical Conf.*, Philadelphia, PA, USA, June 2014, pp. 305-319.

[15] M. Burrows, "The Chubby Lock Service for Loosely-Coupled Distributed Systems," In *Symp. Operating Syst. Des. Implementation*, Seattle, WA, USA, Nov. 2006, pp. 335-350.

[16] J.C. Corbett et al., "Spanner: Google's Globally-Distributed Database," In *Proc. OSDI'12, USENIX Sympos. Oper Syst. Des. Implementation*, Hollywood, CA, USA, Oct. 2012, pp. 251-264.

[17] M. Isard, "Autopilot: Automatic Data Center Management," *Oper. Syst. Rev.*, vol. 41, no. 2, Apr. 2007, pp. 60-67.

[18] B. Calder et al., "Windows Azure Storage: a Highly Available Cloud Storage Service with Strong Consistency," *Proc. ACM Symp. Oper. Syst. Principles*, Cascais, Portugal, Oct. 2011, pp. 143-157.

[19] B.M. Oki and B.H. Liskov, "Viewstamped Replication: A New Primary Copy Method to Support Highly-Available Distributed Systems," *Proc. Annu. ACM Symp. Principles Distributed Comput.*, Toronto, Canada, Aug. 15-17, 1988, pp. 8-17.

[20] F. Junqueira, B. Reed, and M. Serafini, "Zab: High Performance Broadcast for Primary-Backup Systems," In *Proc. USENEX Annu. Techn. Conf.*, Hong Kong, China,

June 27-30, 2010, pp. 245-256.

[21] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," *ACM Trans. Programming Languages Syst.*, vol. 4, no. 3, July 1982, pp. 382-401.

[22] Apache Kafka, Accessed 2017. https://kafka.apache.org/

[23] Apache Zookeeper, Accessed 2017. https://zookeeper.apache.org/

[24] J. Kwon, "Tendermint: Consensus without Mining," 2014, Accessed 2017. https://tendermint.com/static/docs/tendermint.pdf

[25] R3 Corda, Accessed 2017. https://docs.corda.net/

[26] Proof of Elapsed Time of Hyperledger Sawtooth, Accessed 2017. https://sawtooth.hyperledger.org/docs/core/ releases/latest/introduction.html#proof-of-elapsed-time-poet

[27] L. Luu, V. narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A Secure Sharding Protocol for Open Blockchains," In *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, Vienna, Austria, Oct. 2016, pp. 17-30.

[28] Ethereum Sharding, Accessed 2017. https://github.com/ethereum/wiki/wiki/Sharding-FAQ