**<합의알고리즘 이론 및 응용 기말고사 준비>**

**A60031 블록체인전공 홍종화**

**4주차 합의알고리즘 정의 및 개요**

1. **합의 알고리즘 정의 (본인 의견 서술형)**

합의 알고리즘이란 다수의 참여자들이 통일된 의사결정을 하기 위해 사용하는 알고리즘입니다. ‘비트코인: P2P 전자화폐 시스템’에서 제시한 방식에 의하면 과거 분산 시스템 컴퓨팅 분야에서 오래전부터 풀지 못한 비잔티움 장군 문제(Byzantine Generals’ Problem)에 대한 실용적인 해결책으로 등장했습니다. 블록체인에서의 합의 알고리즘은 악의적인 노드가 존재할 수 있는 네트워크 상에서 시스템의 무결성을 보장하고, 모든 노드가 정보를 공유를 한 것의 투명성을 공유하기 위해 존재합니다. 따라서 통일된 의사결정을 내릴 수 있는 중앙이 존재하지 않으므로, 합리적이고 효율적인 의사결정을 내리기 위해 필요합니다.

1. **블록체인과 클라우드의 비교(본인 의견 서술형)**

스크린샷이(가) 표시된 사진

자동 생성된 설명

**6주차 PoW 상세 분석 (1/3)**

1. **HashCash 응용 사례 소개**

1997년 나온 Hashcash는 발신자가 이메일을 보내기 전에 스탬프를 계산하는데 CPU 사용하게 하여 이메일 헤더에 Hashcash 스탬프의 텍스트 인코딩을 추가하게 합니다. 이를 통해 발신 자가 스탬프를 생성하여 전자 메일을 보내기에 스패머일 가능성이 적어집니다. 또한, 계산 비 용을 무시할 정도로 적은 비용으로 스탬프의 유효성을 확인할 수 있습니다. Hashcash를 통해 메시지 당 비용이 거의 없더라도 많은 양의 전자 메일을 보내는 기능에 의존하는 스패머는 수익성이 떨어짐을 알게 되고, 악의적인 행위를 중단하게 됩니다. 이후2002년에 업데이트된 논문에 따르면, 병렬화 할 수 없는 비용 함수가 제공하는 추가 된 DDoS 보호 기능은 미미하다고 이야기합니다. 따라서, 서비스 요청시에 퍼즐을 내어서 자 원 고갈을 막게 하고. 단일 클라이언트로부터 다수 요청시 PoW 문제를 내어 DDoS 공격을 방지합니다. Time-lock puzzle의 사이즈를 줄이기 위해 공개키의 사이즈를 줄이지만 공용 매 개 변수 작업(RSA 암호화에 사용) 자체는 적당히 비싸므로 이 작업은 너무 자주 수행되지 않 습니다. 여기서 발전한 블록체인의 PoW는 해시 기능을 작업 증명 시스템으로 사용합니다. 대부분의 암호 화폐는 SHA-256 해시 기능을 사용하지만 동일한 ASIC 기술을 사용하여 소비자 CPU보 다 3 배 더 빠른 Hashcash Prover를 생성 할 수 있어 스패머의 계산 장애를 줄입니다.

1. **Segwit 정의와 기대효과 설명**

Segwit은 거래와 서명정보를 분리함으로써, 기존 블록에 담을 수 있던 트랜잭션 의 양을 증가시킨 방법입니다. Segwit은 트랜잭션에 있는 서명정보를 블록 외부에 저장하여 트랜잭션ID에 영향을 미치지 않고 서명 및 스크립트를 변경할 수 있도 록 하였습니다. 또한, 블록무게라는 새로운 개념을 도입하여 블록크기의 매시업이 4MB로 제한되고, 기본 트랜잭션 블록의 크기를 1MB로 유지시켰습니다. 이를 통 해 블록에 더 많은 정보를 담을 수 있게 되었으며, 큰 처리량을 감당할 수 있게 되었습니다.

1. **Off-Chain Solution(StateChannel Blockchain) 정의 그리고 사례로 Lightening Network, Raiden Network, Plasma 중 하나를 들어서 설명**

State Channel 은 오프체인 솔루션으로 채널 네트워크의 멀티 허브 라우팅을 통해 작동합니 다. 채널 네트워크는 기본적인 1:1 양방향 채널에서 확장되어 사용자 간의 상태 채널이 유기 적으로 작동하는 상황을 의미합니다. 멀티 허브 라우팅은 채널의 오픈과 관련된 예치 관련 수수료 비용을 최소화하여 오프체인의 효과를 높이는 기법입니다. 상태 채널을 이용한 블록 체인은 대표적으로 라이트닝 네트워크, 라이덴 네트워크, 플라즈마를 들 수 있습니다. - 라이트닝 네트워크 (Lightening Network): 라이트닝 네트워크란 비트코인의 블록체인 위에서 작동하는 두 번째 레이어 솔루션 으로, 비트코인의 블록체인에 기록되는 내용들을 최소화하여, 블록에 담기는 트랜잭 션을 줄이고, 소액으로 자주 거래가 되는 채널을 블록체인이 아닌 다른 레이어에서 처리한 후 결과값만 블록체인에 올리는 방법입니다. 두 당사자가 지불 채널을 온체인에서 개설한 후 오프체인에서 그들이 원하는 만큼 체인 트랜잭션을 실행할 수 있습니다. 트랜잭션 직후 대차대조표의 복사본에 각자 서 명 후 이를 업데이트 하며 지불을 마칠 수 있습니다. 여기서 생성된 스마트 컨트랙트 는 해당 대차대조표의 최종 버전에 따라 그들이 비트코인을 주고 받았다는 사실을 확증합니다. - 라이덴 네트워크 (Raiden Network): 라이덴 네트워크는 이더리움 기반의 ERC-20 토큰을 전송하는데 사용되는 오프체인 솔루션입니다. 안전한 거래를 위해 디지털 서명과 잔액 증명 방식을 통해 체인 연결 보증금으로 완전히 담보가 됩니다. 지불채널을 통해 양도된 금액의 합이 입금된 토큰 을 초과하지 않으면 양방향 전송을 필요한 만큼 할 수 있습니다. - 플라즈마 (Plasma): 플라즈마는 메인체인과 통신하고 상호작용하는 사이드 체인의 기초를 구축하는 것 입니다. 이러한 작업은 블록체인 트리로 작동하도록 설계되며, 트리는 메인체인 위에 수 많은 작은 체인들이 생성될 수 있도록 계층적 배열을 합니다. 이러한 작은 체인을 차일드 체인이라 부릅니다. 플라즈마를 통해 서로 다른 요구를 충족시키는 차일드 체인과 스마트 컨트랙트가 설계되며, 다양한 방법으로 확장성 해결책을 마련할 수 있어 메인체인의 부담을 줄여 주고, 확장성을 취할 수 있습니다.

**7주차 PoW 상세 분석 (2/3)**

1. **비트코인 마이닝 과정 소개. SHA-256과 문제 찾는 것의 원리를 들어서 설명**

비트코인에서 마이닝은 쉽게 10분 간의 거래내역을 가지고 새로운 블록을 생성하는 과정의 의미합니다. 이때, 채굴자는 비트코인이 제시하는 목표 난이도에 부합하는 해시값을 찾아내면 블록이 생성되며, 생성될 때 마다 보상금과 거래 수수료를 받게 됩니다. 마이닝은 모든 노드 가 각 거래마다 독립된 검증을 실시하는 것에서 시작합니다. 어떤 채굴자가 작업 증명에 성 공하면 다른 채굴자들이 검증하여 검증된 거래들을 새로운 블록에 독립적으로 추가하고 다 음 체인에 블록을 연결하는 방식입니다. 이때, 마이닝을 검증하는 것은 모두가 같은 장부를 가지고 있고, 장부가 맞는지 채굴자들이 서로 검증을 하기 때문에 가능합니다. 검증은 정답을 찾은 채굴자에게 받은 논스를 가지고 자기도 해싱을 한 뒤, 그 결과값이 이전 블록의 난이도 목표보다 작다는 사실을 확인함으로 합니다. 마이닝의 기술적인 목표는 블록헤더 안(version, previousblockhash, merklehash, time, bits, nonce 이렇게 6개의 정보)에 있는 난이도 목표보다 낮은 블록 해시값이 나올 수 있게 하는 논스를 SHA-256을 이용해 찾는 방법입니다. 그러나 단순 숫자 대입은 컴퓨팅 파워가 높다면 시간을 줄일 수 있습니다. 따라서 10분 마다 블록을 생성하기 위해 난이도 조정을 하는데 비 트코인은 Hashcash의 방식을 채택하여 구현하고 있습니다. 1) 목표: 2016개의 블록을 생성하는데 2주 가량이 소요되어야 함 2) 작업 방식: 해싱을 통한 목표값 찾기 3) 난이도 조정 방법: 목표치 초과/미달하는 부분만큼 목표값의 난이도가 변경됨 4) 난이도 조정 시점: 매 2016번째 블록마다

1. **GHOST 프로토콜 정의와 Longest Chain Rule과 비교해 포크 대응 방법론 소개**

Ethash는 민주적이고, 비 독점적인 탈중앙 마이닝을 위해 기존 비트코인에서 제작한 ASIC 제작을 어렵도록 설계하였습니다. ASIC의 경우 오로지 마이닝 용도로 사용되며, 에너지 낭비 문제가 있습니다. Ethash의 경우, 계속 확장 변경되는 DAG 파일을 통한 메모리 연산을 요구 합니다. 따라서 미리 알고, 할 수 없는 순차적 메모리 연산으로 인하여 별도 장비 구입 없이 일반적인 PC로 채굴하기 적합하게 설계하였습니다. 그러나 합의 과정을 위해 불필요한 연산 을 통한 에너지 소비는 불가피합니다. 여기에 비트코인의 블록 생성 시간인 10분에서 15초로 줄어들면서 동일한 시간에 생성되는 블록이 많아지게 됩니다. 기존의 Longest Chain Rule을 더욱 이루기 쉬워지며 보안의 위협이 생겼습니다. Ethash에서는 이를 방지하기 위해 고스트 프로토콜을 사용합니다. 고스트 프로토콜이란 Greedy Heaviest Observed Subtree의 약자로 기존 비트코인에서 Longest Chain Rule을 이용한 공격을 방지하고, 메인체인에 인정받지 못한 엉클블록을 인정 함으로 인해 컴퓨터 자원의 낭비와 엉클블록의 증가로 난이도가 낮아져 컴퓨터 파워가 큰 마이너의 영향력이 커짐을 막습니다. 고스트 프로토콜은 블록 생성 시 정상 블록에 최대 2개의 엉클블록까지 보상하여 엉클 블 록의 문제를 해결하는 방법입니다. 기존 Longest Chain Rule의 경우 먼저 가장 긴 체인을 만 들어낸 곳을 캐노니컬 체인이라고 판단합니다. 이에 반해 고스트 프로토콜은 블록의 무게로 유효성을 판단합니다. - 먼저 하나의 블록은 반드시 하나의 부모 블록을 지정하고, 0 또는 그 이상의 엉클 블록을 지정합니다. (최대 2개 까지 지원) - 블록에 포함된 엉클 블록은 다음 3가지 속성을 가지게 됩니다.

- 1) 블록의 K번째 조상의 직접적인 자손이어야 합니다. (2<=K<=7, 7번째 까지 가능). 실제 한 블록이 생성된 후 블록체인에 등록되고 최종적으로 반영된 사실이 확인될 때까지 최대 생성 시점 이후 6개의 블록이 블록체인에 연결될 때 까지 기다립니다. 엉클 블록 역시 앞선 Longest Chain Rule에서 6컨펌 규칙을 따릅니다.

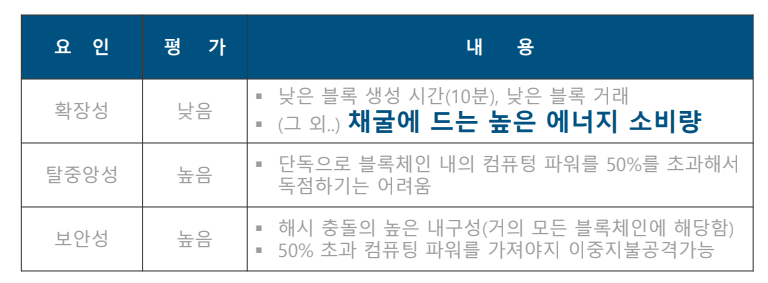
- 2) 블록의 조상이면 안됩니다.

- 3) 엉클 블록은 반드시 유효한 블록 헤더를 가져야 하지만 미리 검증되거나 유효한 블록일 필요는 없습니다.

고스트 기술 문서에 따르면 블록의 마이너 보상은 블록 보상 + 트랜잭션 수수료 + 엉클 블록 1개당 블록 보상의 3.125% 입니다. 엉클 블록의 마이너 보상은 (8-(정상 블록 번호 – 엉클 블록 번호))/8\*블록 보상입니다.

**10주차 PoS 상세 분석 (1/2)**

1. **PoS 등장 배경 서술(본인 의견 서술형)**



1. **PoW와 PoS 나누는 기준 설명(해당 장표의 8쪽 참조 요망)**

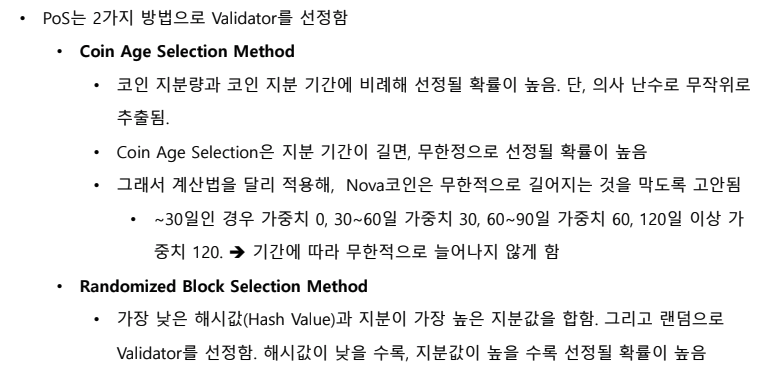
스크린샷이(가) 표시된 사진

자동 생성된 설명

1. **PoS 합의 과정을 소개하고, Validator 선정 알고리즘을 대표적으로 하나를 소개하시오(9쪽과. 35쪽 참조)**

스크린샷이(가) 표시된 사진

자동 생성된 설명



1. **샤딩 기법 도입으로 이더리움 2.0에 예상되는 효과를 기술(해당 장표의 41쪽~45쪽 참조)**

샤딩이라는 것은 본래 데이터베이스(Database)에서 사용하는 방법으로 단일의 논리적 데이터셋을 다수의 데이터베이스에 쪼개고 나누는 방법입니다. 보통 단일 데이터베이스에서 저장하기 힘들다고 판단될 때 사용합니다. 분산 DB로는 Cassandra, Hbase, HDFS, MongoDB 등이 있습니다. 또한, 분할할 때 수직분할, 수평분할로 나눌 수 있습니다. 수직분할의 경우 다른 칼럼을 쪼개 관리하는 방법이며, 수평분할의 경우 테이블이 방대하여 같은 칼럼을 가져가되 DB를 쪼개 관리하는 방법입니다.

블록체인에서의 샤딩은 여러 방면으로 시도 및 적용되려고 합니다. 대표적인 예로 이더리움 샤딩, Zilliqa, Quarkchain 등이 샤딩 솔루션을 적용하고자 노력 중입니다. 사용하는 방식에 따라 샤딩을 하나로 정의할 수 없기 때문에 Zilliqa 팀에서 나눈 기준과 이더리움 샤딩으로 설명드리겠습니다.

1) 네트워크 샤딩(Network Sharding)

네트워크 샤딩을 임의성을 부여하여 네트워크가 무작위로 노드 샘플링을 통해 샤드를 형성하게 합니다(ex: 머클트리의 트랜잭션 트리). 이는 블록단위로 형성하지만 네트워크가 샤드의 구성원에 대해 동의하는지 확인이 필요합니다.

2) 트랜잭션 샤딩(Transaction Sharding)

트랜잭션 샤딩은 트랜잭션 해시의 마지막 몇 비트를 기반으로 샤드를 결정합니다. 이렇게 샤드가 결정되고 블록이 만들어질 때 트랜잭션의 유효성을 확인하는 작업을 합니다. 해당 샤딩의 경우 사용자가 악의적인 경우 동일한 두 입력이지만 출력이 다른 트랜잭션을 생성할 위험이 있습니다. 이를 방지하기위해 유효성 검사 진행 중에 샤드 조각들이 서로 통신을 할 필요성이 있습니다.

3) 상태 샤딩(State Sharding)

해당 방식은 주소기반의 블록체인에 적용되는 방법으로써, 상태가 지정된 블록 체인에서 특정 샤드는 상태의 일부만 유지하는 방법입니다. 상태 샤딩의 경우 Cross-shard communication overhead가 발생하여 교차분할 트랜잭션을 수행하지 못하도록 제한해야하며, 백업 노드를 유지해야합니다. 또한, 네트워크가 재편성이 될 때 일부 동기화가 완료될 때까지 전체 시스템을 사용할 수 없다는 단점이 있습니다.

4) 강의에서 다룬 샤딩(이더리움 샤딩)

- Proposer나 Collator가 되고 싶은 네트워크 참여자는 SMC를 통해 예치금을 예치합니다.

- Collator는 주기적으로 SMC의 상태를 확인하며, 본인이 Collator로 선정이 되어 있는지 확인해야 합니다.

- Collator들은 SMC에 의해 각 샤드체인에 Pseudo-random하게 배정된다. Collator는 Lookahead period 기간 동안 해당 샤드의 이전 기록들을 다운을 받습니다. Collator는 선택된 Proposal을 제안한 Proposer로 부터 Proposal bid를 받게 됩니다.

- Proposer는 트랜잭션을 담은 Proposal을 Collator에게 제출하고, 선택된 Proposal을 제출한 Proposer는 트랜잭션 발송자로부터 트랜잭션 fee를 받습니다.

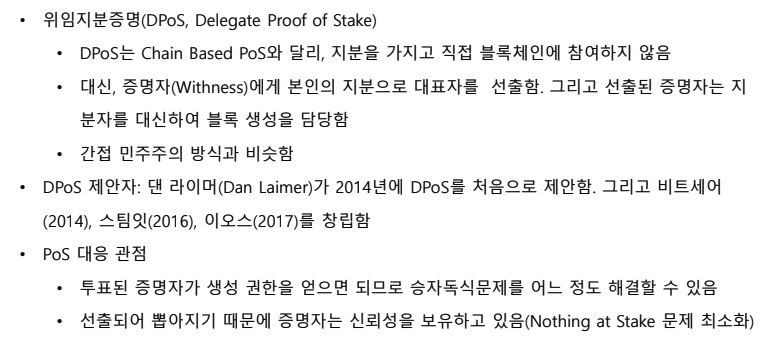
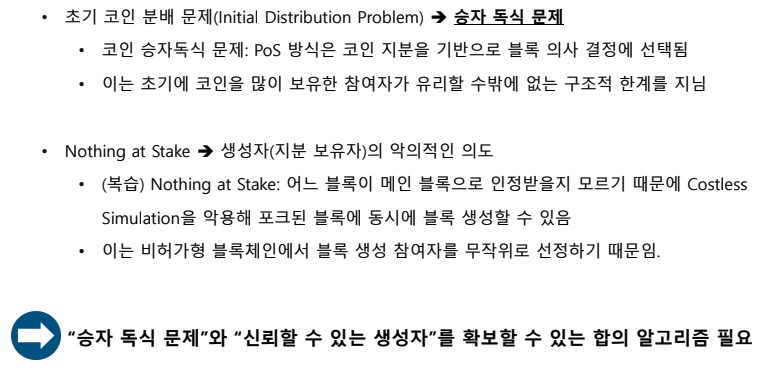
- Collator들은 해당 Proposal에 속한 트랜잭션들이 Valid한지 검증하는 투표를 하게됩니다. 이때 2/3이상의 Collator들이 Proposal에 포함된 트랜잭션이 Valid하다고 찬성하면 해당 Proposal은 유효한 Collation이 됩니다.

- Collator는 add\_header 함수를 호출하여, 투표 이후 새롭게 생성된 Collation header를 SMC로 보내고, SMC를 통해 올라간 Collation header가 메인체인에 연결됩니다.

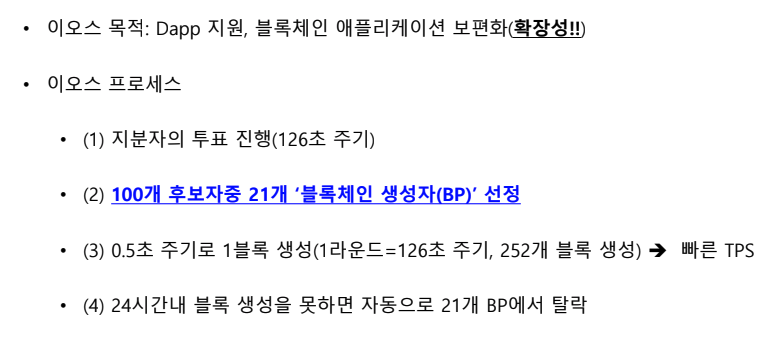
이더리움의 경우 트랜잭션 샤딩과 상태 샤딩의 성격을 같이 가지고 있습니다. 32ETH를 예치한 노드가 검증자로 Beacon Chain에 참여하여, 1024개 샤드에 무작위 배정을 받는 방식으로 개발중입니다. 다만, 난수 생성, 빠른 샤드의 전환, 데이터 가용성 문제, 검증자 간 효율적인 통신 문제 등이 남아있습니다.

**12주차 PoS 상세 분석 (2/2)**

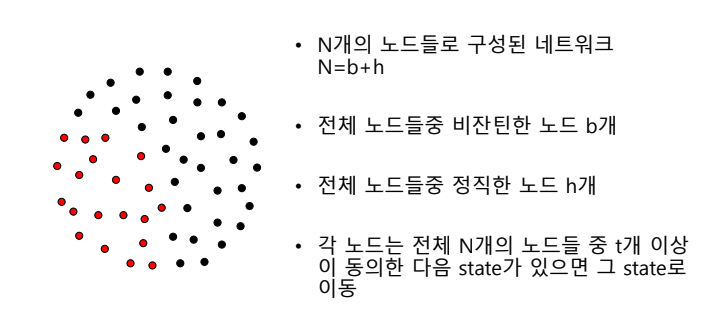
1. **DPoS 등장배경과 정의를 서술하시오(참조자료의 6쪽~7쪽 참조)**



1. **DPoS 대표 사례 하나를 소개하시오(8쪽, 9쪽, 11쪽)**



1. **비잔틴장애허용알고리즘(BFT)에서 3F+1를 계산식을 채택한 이유를 서술하시오(24쪽~32쪽)**



텍스트이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

1. **Practical Byzantium Fault Tolerance 알고리즘을 정의하고, 합의 원리를 간략하게 서술하시오(36~38쪽 참조)**

PBFT 정의

• Validator들이 블록 검증하고, 안전성 수준에 이를 때까지 블록 전파를 기다림

• >3F 혹은 3F+1 특성이 Safety와 Tolerance를 보증함

1) 클라이언트가 서비스 운영(service operation)을 호출하기 위해 primary 노드에게 요청을 전송한다.

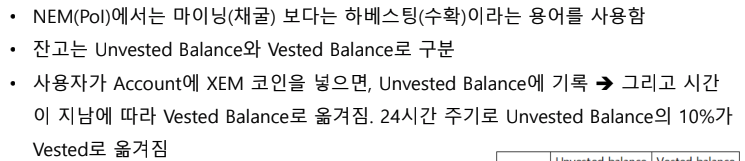
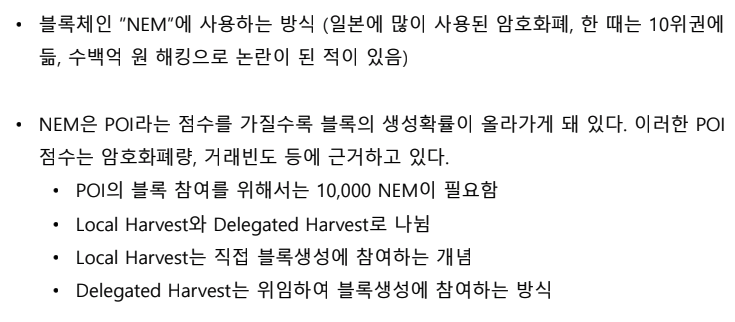
2) Primary 노드는 해당 클라이언트의 메시지 요청을 수집하고 정렬하여 나머지 노드(backups)에게 전송한다.

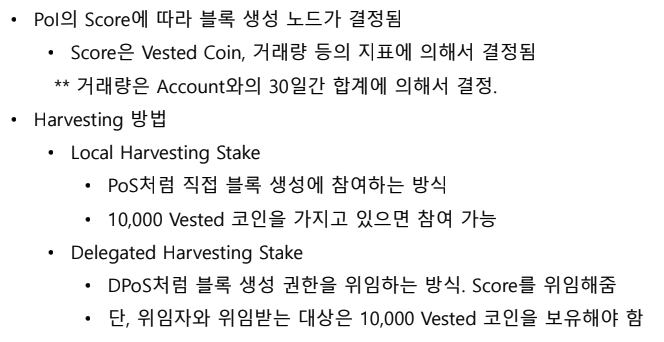
3) 각 노드는 해당 요청사항을 실행하여 검증 절차를 거친다. 모든 검증 작업이 끝나면 해당 메시지에 대한 답변을 클라이언트에게 전송한다

4) 클라이언트는 f+1 개 이상의 동일한 결과를 서로 다른 노드로부터 전달 받으면 해당 요청사항이 문제없이 처리된 것을 확인한다.

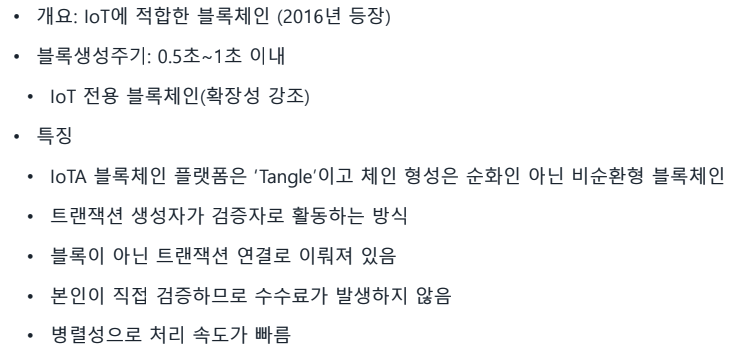
**13 주차 합의 유형 및 설계 고려 사항**

1. **중요증명(PoI)에 관해 기술 하시오(3쪽~5쪽 참조)**





1. **IoTA(Tangle)에 관해 정의하고 특징을 서술하시오(18쪽 참조)**



**14 주차 합의 유형 및 설계 방법론**

1. **합의 알고리즘 설계 시에 고려해야 할 요인이 무엇인지를 자유롭게 서술하시오(본인 의견 서술형), 1단락 참조**

