2017/04/17, 2017 동계 세미나-블록체인

Tendermint: Consensus without Mining

Arthors: Jae Kwon

이 성 범(sungbum@pel.smuc.ac.kr)

상명대학교 프로토콜공학연구실

Contents

- Introduction
- Block Structure
- Validators
- Consensus
- Optimizations
- Conclusion

Introduction

- Bitcoin
  - Peer들은 작업 증명을 통해 블록을 생성
    - 작업을 증명하는데 많은 연산을 함.
  - Double Spending 공격을 방지하기 위해 블록을 6번 확인하는 절차가 필요
    - 올바른 블록으로 인정받기 까지 1시간 소요.
- Bitshare
  - Peer중 대표 Peer를 선출하여 작업 증명하고 블록을 생성
  - 선출된 Peer의 신뢰도를 정확히 측정할 수 없는 문제.
    - Double Spending 공격의 문제가 발생할 수 있음.
- Tendermint
  - Validator들 간의 투표로 블록을 생성하고, Double Spending 공격의 문제를 해결
  - DLS 프로토콜의 수정된 버전 기반
    - ref: C. Dwork, N. Lynch, and L. Stockmeyer, “Consensus in the presence of partial synchrony,” Journal of the ACM, vol. 35, no. 2, pp. 288–323, 1988.

Block Structure

- Block Structure
  - Header Hash
    - H(Header)
  - Validation Hash
    - H(Validation for Block H-1)
  - Transactions Hash
    - H(Transactions)
  - Block H Hash
    - H(Header Hash|Validation Hash|Transactions Hash)

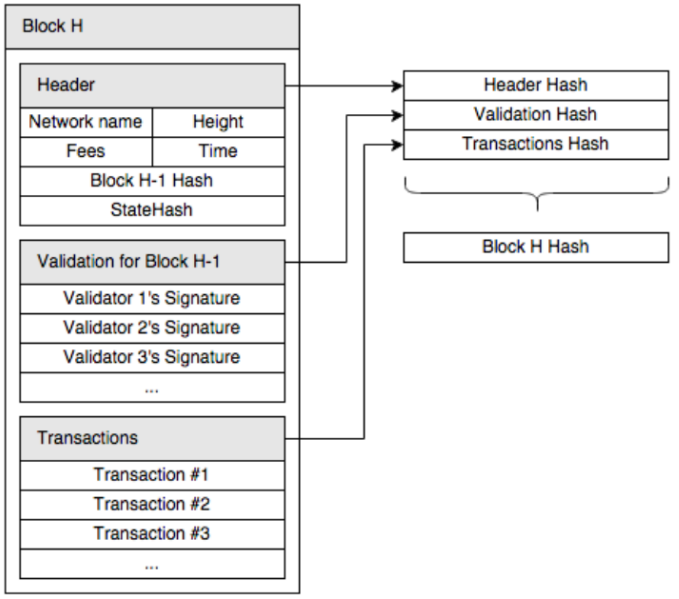


Figure 1: Block structure

- 블록을 통해 Transactions이 유효하다고 할 수 있음.

- ref: R. Merkle, “Protocols for public key cryptosystems,” Proc. 1980 Symposium on Security and Privacy, pp. 122–133, April 1980.

Validators

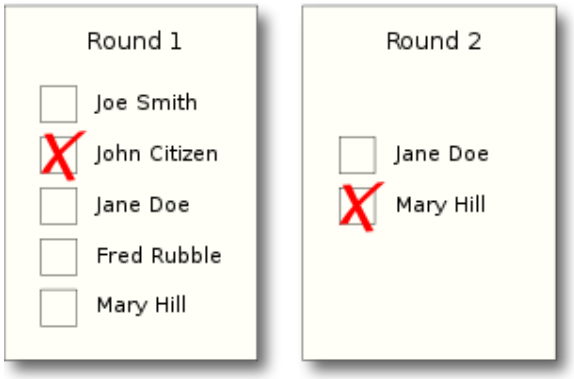
- Validator
  - bond transaction을 Posting하여 bond가 locked 된 계정을 가진 사용자(블록안에 거래를 Posting 한 user)
  - Validator들은 cryptographic signatures, or votes를 Broadcasting하여 블록을 동의함으로써 합의 프로토콜에 참여
  - Validator는 Unbonding Transaction을 Posting 하여 coin을 unlock 할 수 있음.
    - Unlock 후 coin을 사용할 수 있음
  - fork는 동일한 높이의 두 블록이 2/3 다수의 Validator로 각각 서명될 때 발생
    - 정기적인 블록체인 동기화로 해결

Consensus

- On Byzantine Consensus
  - Fischer는 비동기 시스템에서는 합의를 보장할 수 있는 프로토콜은 없다는 것을 보여주었음.
    - FLP(Fischer, Lynch, and Patterson) Impossibility result
    - Bitcoin 에서는 네트워크의 동기화와 시간에 대해 몇가지 가정을 하여 FLP Impossibility result를 해결
  - 이 논문은 Dwork의 논문 Section 4에서 소개하는 알고리즘 2를 기반
    - 4. Partially Synchronous Communication and Synchronous Processors
    - 네트워크가 부분적으로 동기화 되어있다고 가정

Consensus

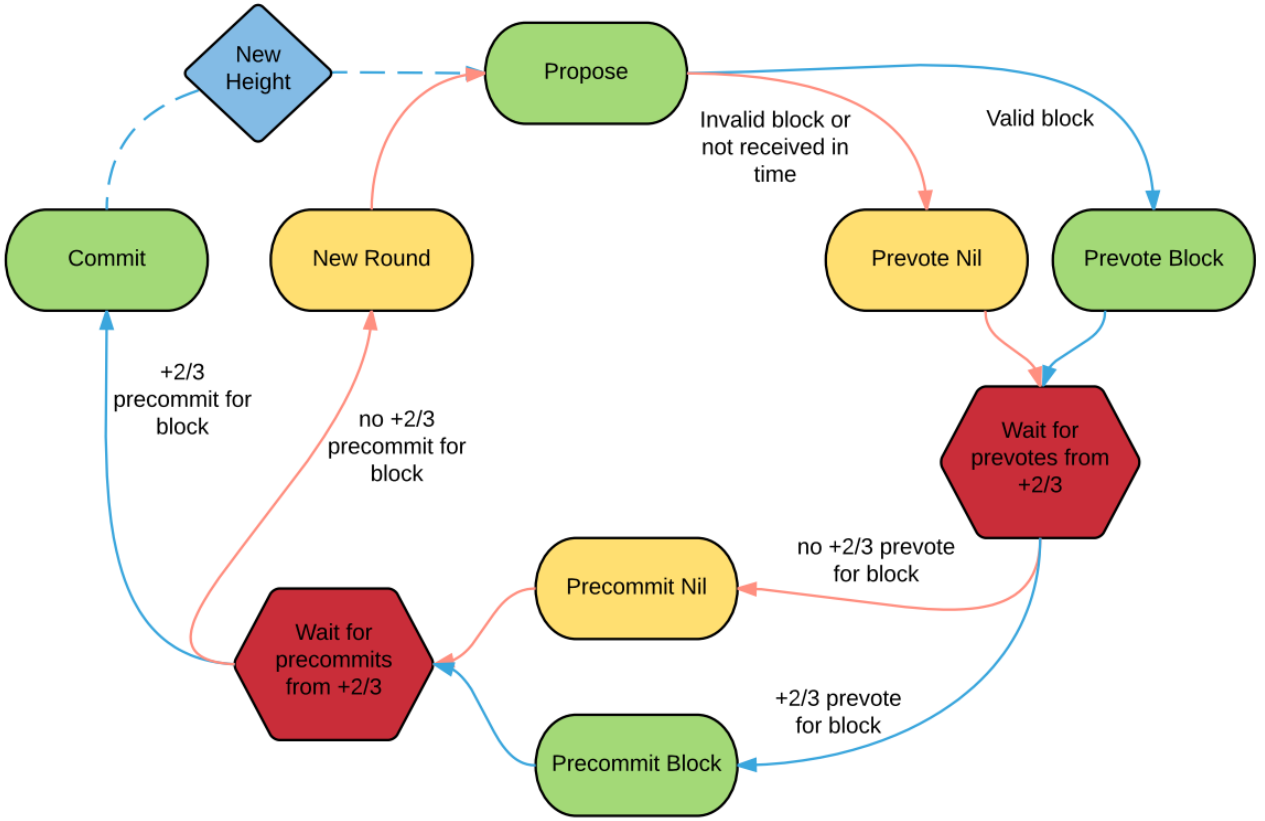
- Algorithm
  - Validator들은 블록 투표에 서명하여 합의 프로세스에 참여
  - 투표의 세가지 단계(블록 생성을 위한 투표 단계)
    - Propose
    - Prevote
    - Precommit
  - 투표의 특별 단계(블록 생성)
    - Commit
    - NewHeight



  - Validator의 2/3가 서명하고 해당 블록에 대한 Broadcast Commit 됐을때 블록이 네트워크에 의해 Commit 됨.
  - 블록체인의 각 Height에서 round-based protocol 이 실행되어 다음 블록을 결정

Consensus

- Algorithm
  - Propose step
  - Prevote step
  - Precommit step
  - Commit step



Consensus

- Algorithm
  - Propose step
    - 제안자는 Proposal Structure를 Broadcast
    - 제안자가 이전 라운드에서 블록에 잠긴 경우 블록을 제안하고 제안서에 Lock 기능을 포함
      - Lock: 동일한 Height의 블록을 커밋하지 못하기 위한 메커니즘
    - 제안 단계에서 모든 노드는 제안을 인접한 노드에게 전달.

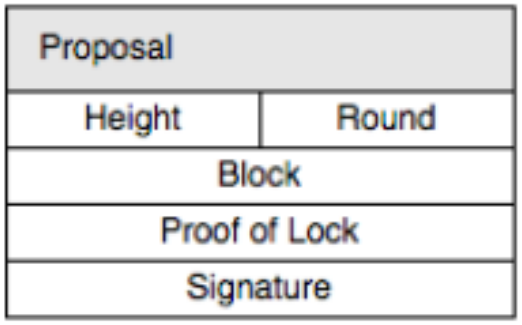
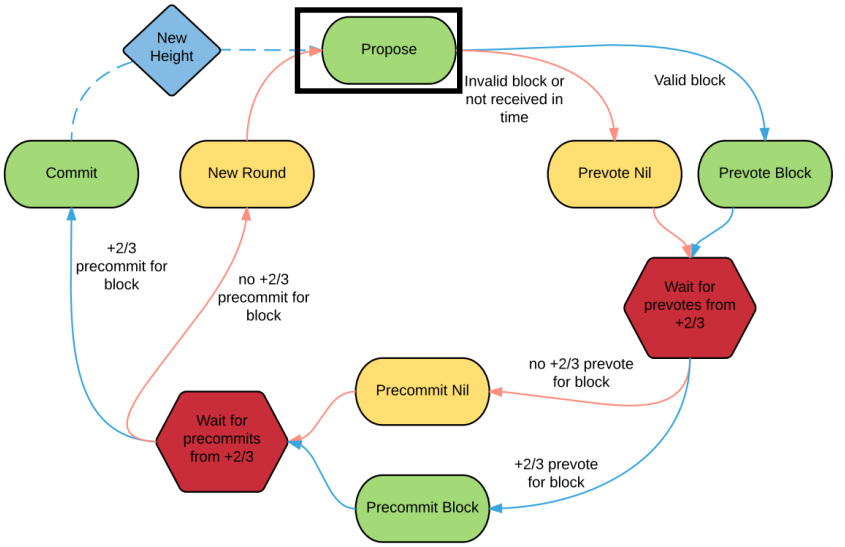
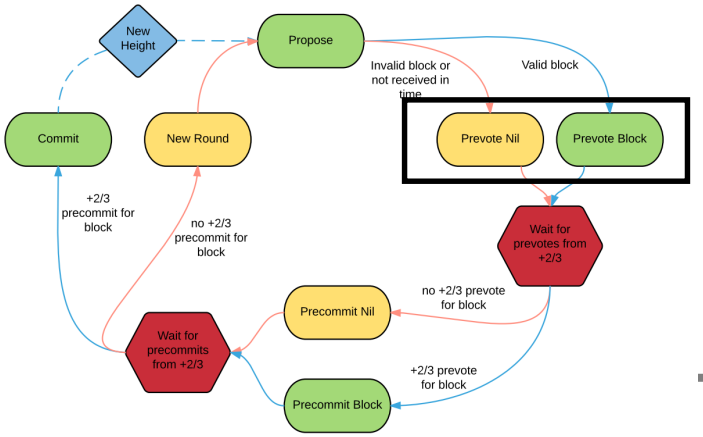


Figure 4: Proposal structure



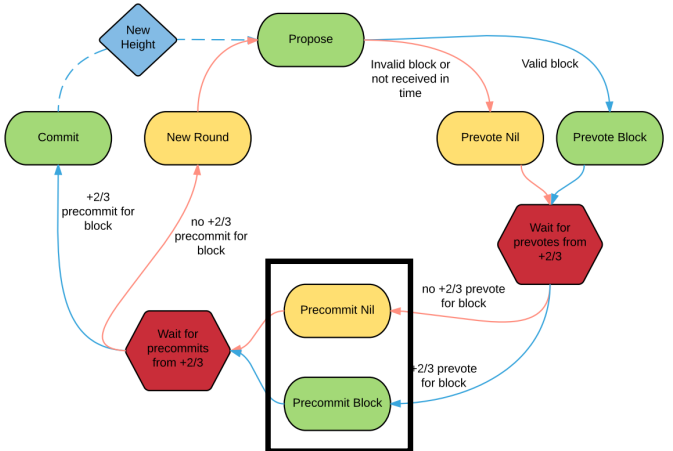
Consensus

- Algorithm
  - Prevote step
    - Propose Lock 인 경우
      - Validator가 이전 라운드에서 제안된 블록이 Lock 되어 있으면 Lock 된 블록에 대해 Prevote를 서명하고 Broadcast 함
    - Propose UnLock 인 경우
      - Validator가 현재 라운드에 대해 서용 가능한 제안을 수신한 경우 제안된 블록에 대해 Prevote를 서명하고 Broadcast 함
    - Propose가 Invalid 한 경우
      - Validator가 no proposal 또는 Invalid 한 제안서를 받은 경우는 special nil prevote(무효)에 서명.



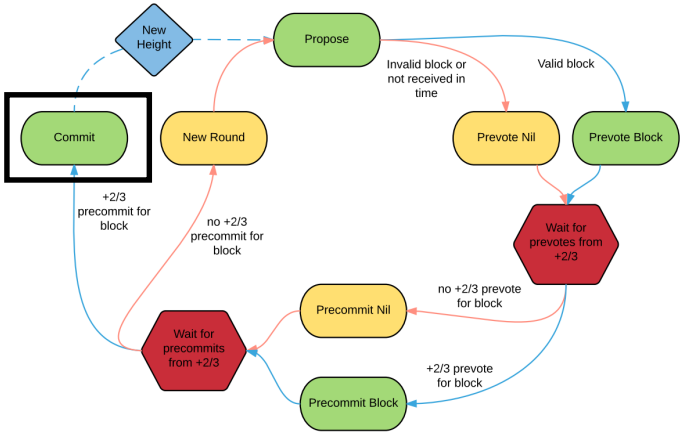
Consensus

- Algorithm
  - Precommit step
    - Validator가 특정 블록에 대해 2/3이상의 prevote를 받은 경우 validator는 해당 블록에 대한 precommit에 서명하고 broadcast
    - 노드가 특정 블록에 대해 2/3이상의 prevotes를 받지 못하면 서명을 하지 않음
    - Precommit단계에서 모든 노드는 모든 인접한 노드에게 라운드에 대한 모든 Precommit을 시도함.
    - 노드가 특정 블록에 대해 2/3이상의 Precommit을 수신한 경우 노드는 Commit 단계에 진입



Consensus

- Algorithm
  - Commit Step
    - 라운드를 마무리하는 단계로 두가지 조건이 만족되어야 함
      - 노드는 네트워크에 의해 Commit 된 블록을 수신해야 함
      - 노드는 네트워크에 의해 미리 결정된 블록에 대해 최소한 commit의 2/3을 수신할 때 까지 대기해야 함
      - 두 조건이 충족되면 노드는 CommitTime을 현재 시간으로 설정하고 NewHeight 단계로 전환
        - NewHeight단계: Commit을 추가하는 단계
    - 노드는 CommitTime이 지난 일정 기간 동안 유지됨



Consensus

- Proof of Safety
  - Byzantine voting power가 1/3 이하인 Validator가 블록 B를 결정한 경우, 라운드에서 블록 B를 커밋하려고 함
  - 2/3이상의 Precommit을 받았지만, Commit 하기 위해 1/3 이하의 vote 밖에 받지 못했기 때문에 블록은 Lock.
- Proof of Liveness
  - 컨센서스 프로세스가 교착 상태에 빠질 수 있는 방법
    - 두개의 블록이 여러 라운드에서 Validator에 의해 잠겨져 있는 경우
      - 잠금을 해제하여 합의과정을 계속 진행
- Cooperation
  - Validator는 블록 수수료를 서명을 한 Validator들과 배분

Optimizations

- Sparse Signature Set
  - 설명된 프로토콜은 이론적으로 가능하지만, 실제로는 계산, 저장 및 네트워크 제한사항을 고려해야 함
  - Validator의 서명을 모든 블록에 저장하는 것은 어려움.
    - Describe sparse signature set as DMMV
- Handling Validator timeouts
  - 네트워크는 Validator의 상태 변경에 대해 적응 해야함
  - 오프라인 된 Validator는 bond transaction을 다시 제출하여 활성화
- Faster Block Propagation
  - TODO: Describe lib-swift used for broadcasting blocks
    - ref: Riccardo Petrocco, Johan Pouwelse, and Dick HJ Epema. “Performance analysis of the libswift p2p streaming protocol”. In: Peer-to-Peer Computing (P2P), 2012 IEEE 12th International Conference on. IEEE. 2012, pp. 103–114.

Conclusion

- Tendermint is awesome. The Future is now

8. Conclusion

Tendermint is awesome. The future is now.

- Discussion
  - Gossip Network 취약점
    - 네트워크 구성 및 프로토콜을 파악하면, eclipse attack 이 가능?
      - 투표의 내용을 조작.
  - 사용자에 대한 서명 검증 방법 공격.
    - 한명이 다수의 위조된 서명으로 투표를 진행할 수 있을까?

감사합니다!

(sungbum@pel.smuc.ac.kr)