합의 알고리즘에 따른 블록체인 기반 투표 시스템 성능 평가*

예지훈○, 김민우, 노태완, 김형백, 장두성†
서강대학교 컴퓨터공학과
{ye422, minwoo2246, heize0502, gudqor1, dschang}@sogang.ac.kr

Performance Evaluation of a Blockchain-Based Voting System According to Consensus Algorithms

Jihun Ye○, Minwoo Kim, Taewan Noh, Hyeongbaek Kim, Duseong Chang†
Department of Computer Science and Engineering, Sogang University

요 약

본 연구는 블록체인 기반 투표 시스템을 설계하고 합의 알고리즘별 성능 차이를 분석한다. 블록체인은 분산 원장 구조를 통해 데이터 위·변조를 방지하고 투명성을 확보할 수 있으나 합의 알고리즘에 따라 처리 속도(TPS)와 확정 시간이 달라져 실용화에 제약이 존재한다. 이를 검증하기 위해 GoQuorum 기반 프라이빗 블록체인 환경을 구축하고 IBFT, QBFT, RAFT 세 가지 합의 알고리즘을 동일한 조건에서 비교하였다. 실험은 7개의 검증 노드와 1개의 RPC 노드로 구성된 네트워크에서 수행되었으며, 스마트컨트랙트를 이용해 투표 트랜잭션을 자동 처리하고 각 합의 구조의 성능을 정량적으로 측정하였다. 실험 결과, RAFT는 평균 확정 시간이 약 0.245초로 가장 빠르고 안정적인 응답성을 보였으며, QBFT는 RAFT 대비 약 1.5배 높은 처리량을 기록해 대규모 트랜잭션 환경에서 우수한 효율을 보였다. 반면 IBFT는 높은 통신 복잡도로 인해 상대적으로 낮은 성능을 나타냈다.

1. 서론

현대 사회에서 투표는 민주주의를 유지하는 핵심 수단이지만 기존의 온라인 투표 시스템은 중앙 서버에 의존함으로써 조작 가능성과 단일 장애점을 내포한다. 이러한 구조적 한계는 투표 결과의 신뢰성과 투명성을 저하시켜 기술적 불신과 사회적 갈등을 초래할 수 있다.

최근 블록체인 기술은 이러한 문제를 해결할 수 있는 대안으로 주목받고 있으며 분산 원장 구조를 통해 데이터 위·변조를 방지하고 모든 참여자가 동일한 원장을 검증함으로써 높은 수준의 신뢰성을 확보할 수 있다.

그러나 블록체인 기반 투표 시스템의 실용화에는 여전히 해결해야 할 과제가 존재한다. 특히 합의 알고리즘에 따라 TPS와 확정 시간이 상이하게 나타나며, 이는 실제 서비스 환경에서 시스템의 성능과 사용성을 결정짓는 핵심 요인이다.

본 연구는 이러한 성능을 검증하기 위해 GoQuorum [1] 기반의 프라이빗 블록체인 환경에서 서로 다른 합의 알고리즘을 적용하고 처리 속도와 블록 확정 시간을 중심으로 성능을 비교·분석하여 투표 시스템에 가장 적합한 합의 구조를 도출한다.

* 본 연구는 2025년 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학사업 지원을 받아 수행되었음 (2024-0-00043)
† Corresponding author

2. 연구 배경

2.1. 블록체인 개요

블록체인은 중앙 관리자 없이 여러 참여자가 동일한 원장을 분산 저장·검증하는 분산 원장 기술이다. 각 블록은 이전 블록의 해시를 포함해 체인 구조로 연결되어 데이터 위·변조를 방지한다. 이러한 특성은 중앙 서버에 의존하는 기존 시스템의 조작 가능성 및 단일 장애점 문제를 해결할 수 있다.

스마트컨트랙트는 블록체인에 배포되어 조건이 충족되면 자동으로 실행되는 프로그램으로, 중개자 없이 계약을 수행한다. Solidity 등의 언어로 작성되어 이더리움 가상 머신에서 동작하며 모든 실행 내역은 블록체인에 영구히 기록되어 위·변조가 불가능하다. 이는 투표·금융·공급망 등 다양한 서비스를 탈중앙화 형태로 구현할 수 있게 하여 블록체인의 투명성과 신뢰성을 응용 서비스 수준으로 확장하는 핵심 요소로 작동한다.

블록체인 네트워크는 퍼블릭 블록체인과 프라이빗 블록체인으로 나눌 수 있다. 퍼블릭 블록체인은 누구나 참여 가능하지만 트랜잭션 처리 속도가 느리다. 이에 반해,

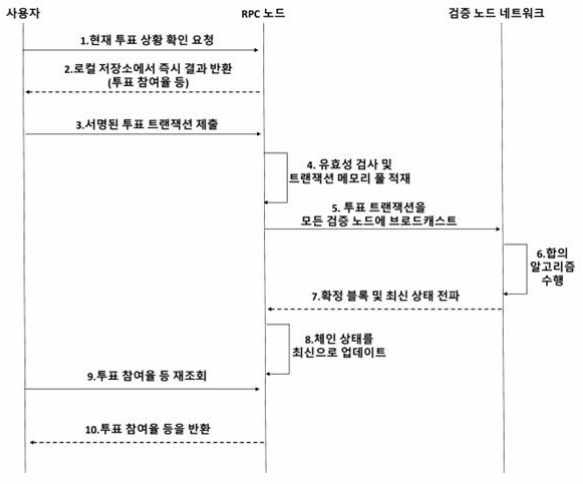

그림 1. 블록체인 기반 투표 시스템 동작 과정

프라이빗 블록체인은 승인된 참여자만 접근 가능하여 효율적이다 [2].

2.2. 합의 알고리즘 개요

합의 알고리즘은 분산 네트워크의 모든 노드가 동일한 원장 상태에 도달하도록 하는 핵심 절차로, 블록체인의 신뢰성과 성능을 결정한다. 퍼블릭 블록체인은 누구나 참여할 수 있어 보안성을 중시한 PoW [2], PoS [2]를 사용하지만, 프라이빗 블록체인은 신뢰된 노드 간 효율성과 확정 시간이 우선된다.

합의 구조는 허용하는 장애 유형에 따라 BFT(Byzantine Fault Tolerance)와 CFT(Crash Fault Tolerance)로 나뉜다. BFT는 악의적 노드의 존재를 가정하고 정합성을 유지하며 CFT는 단순 장애나 응답 불능만을 고려한다. 프라이빗 환경에서 주로 사용되는 알고리즘은 다음과 같다.

- IBFT [3]: BFT 기반으로, 리더 제안 블록을 검증 노드들이 세 단계 투표(Pre-prepare, Prepare, Commit)로 승인하며 3분의 2 이상 동의 시 확정된다. 안정적이지만 통신 부하가 크다.
- QBFT [4]: IBFT의 개선형으로, 메시지 교환과 라운드 전환 절차를 단순화해 처리 효율과 예측 가능한 확정 시간을 확보했다.
- RAFT [5]: CFT 기반 리더 중심 방식으로, 과반수 노드 응답만으로 합의가 완료된다. 효율적이지만 악의적 노드에 대한 방어 능력은 제한적이다.

3. 연구 방법론

3.1. 시스템 설계

본 시스템은 그림 1과 같이 사용자 계층, RPC 노드 계층, 검증 노드 네트워크 계층으로 구성된다.

사용자 계층은 DApp을 통해 투표 참여 및 결과 조회를 담당한다. 사용자는 블록체인과 직접 통신하지 않고 RPC 노드를 매개로 요청을 주고받는다. 투표 시 사용자의 개인키로 트랜잭션이 서명되어 RPC 노드에 전송되며 안전한 참여가 보장된다.

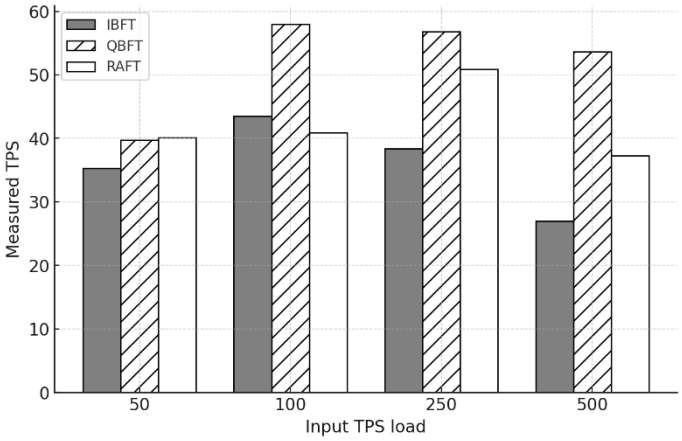

그림 2. 합의 알고리즘별 TPS 성능 비교

RPC 노드 계층은 사용자의 요청을 검증하고 검증 노드 네트워크로 전달하는 중간 처리 계층이다. 조회 요청은 노드가 보유한 로컬 체인 데이터를 이용해 즉시 응답하며 투표 트랜잭션은 서명·형식·권한 검증 후 메모리 풀에 등록된다. 이후 유효한 트랜잭션만을 검증 노드에 전파해 합의 절차가 시작되도록 한다. 블록이 확정되면 RPC 노드는 새 블록 정보를 반영해 체인 상태를 동기화한다.

검증 노드 네트워크 계층은 합의 알고리즘(IBFT, QBFT, RAFT) 투표를 거쳐 일정 비율 이상 승인되면 체인에 추가되고 확정된 블록은 RPC 노드로 전파되어 전체 네트워크 상태가 갱신된다.

이 과정을 반복함으로써 모든 투표 기록은 위·변조가 불가능한 형태로 저장되며 중앙 관리자의 개입 없이 신뢰성과 투명성을 갖춘 투표 환경을 구현한다.

3.2. 실험 설정

본 실험에서는 프라이빗 블록체인 네트워크를 구축하여 IBFT, QBFT, RAFT 세 가지 합의 알고리즘의 성능을 정량적으로 비교 분석하였다.

ConsenSys Quorum Dev Quickstart [6]의 Docker Compose 환경을 이용하여 7개의 검증 노드와 1개의 RPC 노드로 구성된 프라이빗 네트워크를 구축하고 사용자의 투표를 저장하는 스마트컨트랙트를 배포하였다. 블록 생성 주기는 IBFT와 QBFT는 2초, RAFT는 300밀리초로 설정하였다.

부하 조건은 50, 100, 250, 500 TPS로 설정하였으며, 각 조건당 5회씩 반복하여 총 60회의 테스트를 수행하였다. 측정 지표는 초당 처리된 투표 트랜잭션 수(TPS)와 트랜잭션 전송부터 블록 포함까지 소요된 확정 시간이다.

성능 측정은 파이썬 스크립트를 통해 수행하였다. 이 스크립트는 Web3.py 라이브러리를 사용하여 블록체인 RPC 노드와 통신하며 실험 과정을 자동화한다. 트랜잭션 전송 후에는 각 트랜잭션의 전송 시각과 블록 포함 확인 시각을 기록해 TPS와 확정 시간을 측정한다.

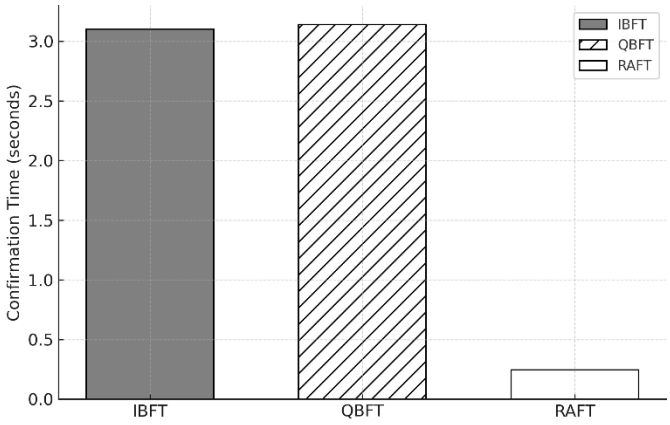

그림 3. 합의 알고리즘별 평균 확정 시간

4. 실험 결과

그림 2는 TPS 실험 결과를 나타낸다. 부하 수준이 낮을 때(50 TPS 이하)에는 RAFT와 QBFT가 유사한 처리량을 보였으며 RAFT는 약 40 TPS 수준에서 안정적으로 포화되는 양상을 나타냈다. 반면 QBFT는 부하가 증가함에 따라 처리량이 점진적으로 향상되어 100 TPS 부하에서 약 58 TPS를 기록하며 RAFT 대비 약 1.4배 높은 처리 효율을 보였다. IBFT는 전 구간에서 가장 낮은 처리량을 기록하였으며 이는 다중 서명 검증 및 블록 제안자 교체 시 통신 오버헤드가 추가적으로 발생하기 때문으로 해석된다. 결과적으로, QBFT는 다수의 검증 노드 간 메시지 교환에도 불구하고 효율적인 파이프라인 합의 구조를 통해 고부하 상황에서 우수한 처리량을 보였으며 RAFT는 단일 리더 기반 합의 덕분에 저부하 환경에서 짧은 지연과 안정적인 처리 성능을 확보한 것으로 분석된다.

그림 3은 확정 시간 측면의 성능을 비교한 결과이다. RAFT는 평균 확정 시간이 약 0.245초로, 세 알고리즘 중 가장 빠른 응답성을 보였다. 이는 리더 노드가 블록을 생성함과 동시에 커밋을 수행하는 구조적 특성에 기인하며 트랜잭션이 블록 생성 직전에 도착할 경우 즉시 확정되는 경우가 많아 평균 확정 시간이 블록 생성 주기보다 더 짧게 측정되는 현상도 나타났다.

반면 QBFT와 IBFT는 평균 약 3초 이상의 확정 시간을 기록하였는데, 이는 블록 제안 이후 다수 노드 간의 합의 과정을 거쳐야 최종 확정이 이루어지는 BFT 구조 때문으로 분석된다.

5. 결론 및 향후 연구

종합하면, 제안된 블록체인 기반 투표 시스템에서 RAFT 합의 알고리즘은 확정 시간이 매우 짧고 TPS도 일정 수준으로 확보되어 실시간 투표 현황 표시나 소규모 내부 투표처럼 사용자 응답 지연에 민감한 환경에 적합하다. 반면 QBFT 합의 알고리즘은 확정 시간은 상대적으로 길지만 최대 처리량이 RAFT 대비 약 1.4배 높아, 대규모 트랜잭션이 발생하는 공공 선거나 기업 단위의 대규모 온라인 투표 환경에서 더 안정적인 성능을 제공할 수 있다. 따라서 합의 구조의 선택은 투표 규모, 노드 간 신뢰 수준, 요구되는 확정 시간에 따라 달라질 수 있으며 신뢰할 수 있는 노드만 참여하는 내부 조직·기업 내 투표 환경에서는 RAFT가, 참여자 수가 많고 결과의 무결성이 특히 중요한 공공 선거나 위탁 선거 환경에서는 QBFT가 더 적합한 선택이 될 수 있다. 이처럼 환경에 맞는 합의 알고리즘을 채택함으로써 시스템은 응답 지연과 처리량, 신뢰성을 균형 있게 확보할 수 있다.

향후 연구에서는 합의 알고리즘의 환경별 선택 기준을 보다 정량적으로 정의하기 위해 노드 수, 투표 참여 패턴, 네트워크 지연·패킷 손실률 및 네트워크 트래픽·하드웨어 성능과 같은 외부 요인을 체계적으로 변화시키며 성능을 분석할 필요가 있다. 또한 본 논문에서 다루지 않은 악의적 노드나 네트워크 공격 시나리오를 포함하여 RAFT와 QBFT의 안전성을 평가함으로써 단순 성능 지표를 넘어 보안 요구 수준에 따른 합의 알고리즘 선택 가이드라인을 제시할 수 있을 것이다. 나아가 실제 공공 선거 또는 기업 내부 투표 시나리오에 본 시스템을 시범 적용하고 사용자 경험 및 운영 편의성 측면을 추가로 검증함으로써 제안 시스템의 실용성을 보다 구체적으로 입증하는 연구가 필요하다.

참고문헌

[1] ConsenSys, Quorum Official GitHub Repository, Available: https://github.com/Consensys/quorum

[2] 민연아, 임동균, "NFT 거래 안정성을 고려한 합의 알고리즘 성능 분석," 한국인터넷방송통신학회논문지, 제22권, 제2호, 2022.

[3] ConsenSys, GoQuorum Documentation: IBFT Consensus Protocol, Available: https://docs.goquorum.consensys.io/configure-and-manage/configure/consensus-protocols/ibft

[4] ConsenSys, GoQuorum Documentation: QBFT Consensus Protocol, Available: https://docs.goquorum.consensys.io/configure-and-manage/configure/consensus-protocols/qbft

[5] ConsenSys, GoQuorum Documentation: RAFT Consensus Protocol, Available: https://docs.goquorum.consensys.io/configure-and-manage/configure/consensus-protocols/raft

[6] ConsenSys, Quorum Official GitHub Repository, Available: https://github.com/Consensys/quorum-dev-quickstart