LFT: Byzantine Fault Tolerance를 지원하는 경량화된 고성능 합의 알고리즘

theloop

June 23, 2017

Abstract

최초의 블록체인 구현 서비스인 비트코인은 작업증명 (Proof of Work) 알고리즘을 이용하여 전 세계 규모의 네트워크에서 거래장부에 대한 합의를 이루었다. 그러나 비트코인에서 사용한 작업증명 알고리즘은 낮은 처리 속도와 비효율적인 에너지 사용, 부분적인 네트워크 분기가 생기는 문제 때문에 거래의 효율성이 중요하거나 즉각적인 거래 완료를 요구하는 환경에서는 사용이 어려웠다. 이러한 문제를 해결하기 위해 Tendermint, Hyperledger Fabric 등의 블록체인은 전통적인 상태 기계 복제 프로토콜인 PBFT 기반 합의 알고리즘을 도입하고 있다. 그러나 PBFT 기반 합의 프로토콜은 많은 통신량으로 인한 성능저하가 일어날 수 있다. LFT는 비잔틴 문제를 해결하기 위한 Raft 알고리즘인 Tangora를 블록체인에 적합하게 개선한 합의 알고리즘으로 PBFT의 많은 통신량 문제를 개선하였고 기존 알고리즘의 복잡한 리더 선정 알고리즘을 단순화하고 리더의 요청 무시 문제를 최소한 경량화된 고성능 합의 알고리즘이다.

1 서론

비트코인은 거래 당사자가 신뢰하는 제3의 신뢰 기관 없이 사용자 간 직접 송금이 가능한 암호화 화폐이다.[1] 비트코인은 블록체인 데이터 구조를 이용하여 데이터 위변조 공격을 즉각 감지할 수 있으며 작업 증명(Proof-of-Work)합의 알고리즘을 이용하여 세계적인 규모의 네트워크에서 분산합의가 가능하도록 하였다.[1] 블록체인은 어떠한 데이터도 복제할 수 있고 신뢰할 수 없는 디지털 환경에서 신뢰 네트워크를 구성할 수 있게 하였고 이를통해 암호화 화폐 분야뿐 아니라 은행 간 거래, 본인 인증, 보험업 등 다양한 분야에 혁신을 이뤄낼 것으로 기대된다.

그러나 비트코인 블록체인에서 사용하는 작업 증명 알고리즘은 긴 처리 시간과 낮은 처리량, 에너지 낭비, 부분적인 네트워크 분기 문제가 있어 요청의 즉각적인 완료를 요구하는 다양한 환경에서는 활용의 한계가 있다.[2] 특히 신원이 확인된 참여가간 구축되는 허가형 블록체인의 경우 그 특성상 사용자에게 즉각적인 서비스를 제공하는 경우가 많아 작업증명 알고리즘을 사용할 수 없는 경우가 많다.

이러한 블록체인 분산합의 알고리즘의 한계를 극복하기 위하여 Tendermint, Hyperledger Fabric 등의 블록체인은 전통적인 상태 기계 복제에 사용하는 합의 알고리즘인 PBFT(Practical Byzantine Fault Tolerance)를 블록체인에 적합하게 개량하여 사용한다.[3][4][5][6] 기대되는 행동을 하지 않는 비잔틴 노드의 갯수가 f라고 할때 PBFT는 3f+1 이상의 노드에서 비잔틴 장애를 극복할 수 있는 합의 알고리즘으로

비잔틴 노드에 의한 네트워크 분기 시도와, 네트워크 정지 시도를 막기 위하여 네트워크의 모든 노드들에 2번에 거쳐 합의 데이터를 전송한다. Tendermint와 같은 선행 연구들은 PBFT를 이용하여 비트코인으로 대표되는 작업증명 알고리즘을 사용하는 블록체인 시스템의 문제였던 높은 지연 시간과 낮은 처리량 문제,부분 블록체인 분기 문제를 해결하였다.[7]

그러나 이러한 PBFT기반 합의 알고리즘은 비잔틴 장애를 극복하기 위하여 합의 결과를 네트워크의 모든 노드에 2번에 걸쳐 전파하기 때문에 통신량이 많아지는 문제가 존재한다. 그래서 비잔틴 장애가 없는 상황에서 상태 기계 복제를 위해 사용하던 프로토콜인 Paxos와 Raft를 비잔틴 장애를 극복하도록 개량하여 만든 알고리즘을 도입하기 시작하였다. [8][9] Tangaroa는 Raft 알고리즘의 투표 메시지를 모든 노드에 전송하는 방식과 리더 노드의 장애를 감지하고 리더 변경하는 메커니즘을 통해 비잔틴 장애를 극복할 수 있는 Raft 알고리즘을 개선하였으며, Juno와 ScalableBFT와 같은 프로젝트에서는 이러한 Tangaroa는 알고리즘을 블록체인에 맞게 개선하여 사용한다.[10][11][12]

LFT는 비잔틴 장애를 극복할 수 있는 블록체인을 위한 고성능 합의 알고리즘으로써 비잔틴 장애 극복 Raft 알고리즘인 Tangora에 기반을 둔 합의 알고리즘이다. Tangaroa 합의 알고리즘의 복잡한 리더 선정 알고리즘 문제와 리더 노드에 의한 특정 노드 요청 거부 문제를 해결하기 위해 Spinning 기법을 적용하였고 리더 선정 메시지와 합의 투표 메시지를 통합시켜 알고리즘을 단순화하였다. 또한, 블록에 합의 내역을 저장하는 방식을 통하여서 전송 불량으로 인한 부분 분기가 일어나지 않도록 하였다. 또한 특정 거래에 대하여 어떤 노드가 합의했는지 확인이 가능하여 합의에 대한 책임소재를 가려야 하는 환경에서도 사용이 가능하다. [13]

2 관련 연구

본 장에서는 현재 다른 블록체인에서 이용하고 있는 전통적인 상태 기계 복제 알고리즘 기반의 블록체인 합의 알고리즘을 소개하고, 해당 알고리즘과 LFT의 차이점에 관해 설명한다. 2.1장에서는 PBFT기반 블록체인 합의 알고리즘인 Tendermint와 2.2장에서는 Raft 기반 블록체인 합의 알고리즘인 ScalableBFT에 대해서 설명한다. [6][3][9][12]

2.1 Tendermint

[3] Tendermint는 공개 블록체인과 사설 블록체인을 지원하는 블록체인으로써 DPoS (Delegated Proof-of-Stake)에서 아이디어를 얻어 전통적인 PBFT를 DPoS형태로 개량하여 참여가 자유로운 공개 블록체인에서 돌아갈 수 있게 한 PBFT 방식의 알고리즘이다. [14] [6]

Tendermint는 전통적인 상태 기계 복제 알고리즘이 노드 수에 따라 다수 노드의 동의로 장애를 극복하는 것에 반해 지분을 이용해서 비잔틴 장애를 극복한다. 이는 기존 지분증명(Proof-of-Stake)과 유사한 게임이론을 통해 공격을 방지하는 방식이다. 해당 블록체인 네트워크에 많은 지분을 가진 노드는 네트워크가 파괴되는 상황보다 네트워크가 유지되어 블록체인이 안정적으로 운영하는 것이 이득이 된다.[15] 또한 Tendermint는 합의 시 자신의 지분을 네트워크에 증거로 맡기고 이중 지불 공격과 같은 블록체인 네트워크를 파괴하기 위한 공격을 시도하면 해당 지분을 전부다 뺏는 방식으로 블록체인 네트워크에 대한 공격자의 공격 시도에 기존 블록체인 시스템이

아무런 처벌을 하지 않는 문제에 대응하고 있다.(Nothing at Stake Problem)

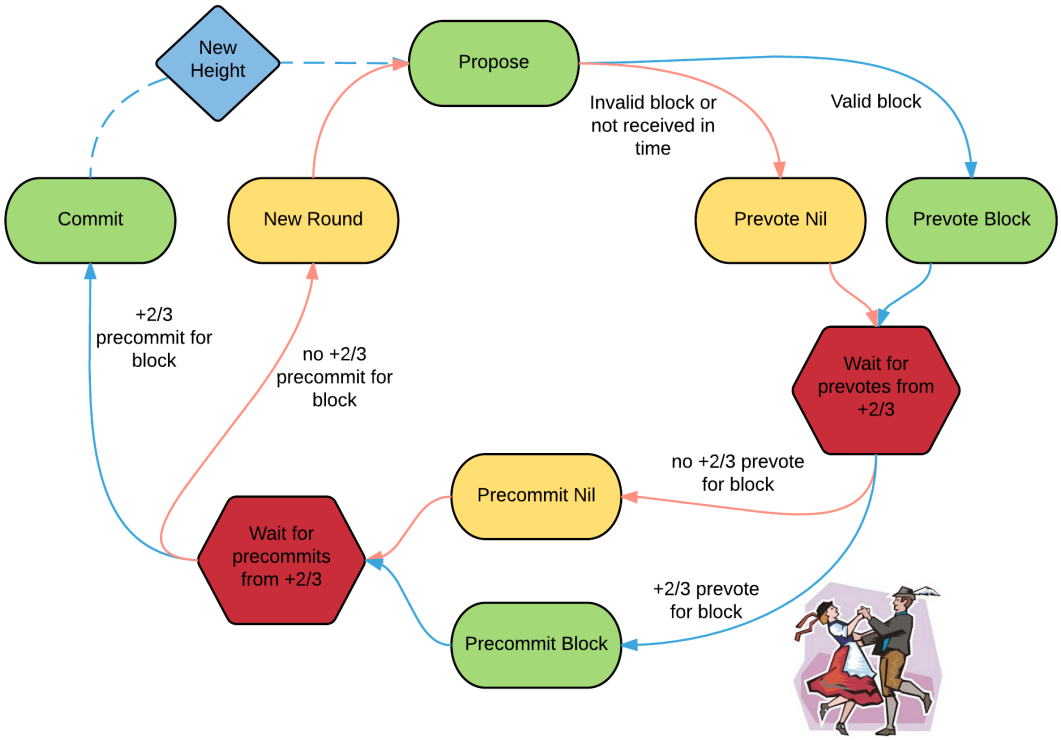

Figure 1: Tendermint 합의 과정[3]

그림 1는 텐더민트의 합의 과정을 나타낸 그림이다. Tendermint는 Propose, Prevote, Precommit 단계를 통해 네트워크에 메시지를 전달하고 Commit 단계를 통해 새로운 블록을 자신의 블록체인에 연결하는 방식으로 전체 과정이 이루어진다. Propose 단계에서는 리더가 블록을 생성하여 다른 노드들에게 전파하고, 각 노드는 Prevote 단계에서 블록을 검증해 Prevote 메시지를 전파한다. 각 노드가 2/3이상의 Prevote 메시지를 받았으면 Precommit 메시지를 모든 노드에 전파하고 2/3 이상의 Precommit 메시지가 공유된 블록은 각 노드가 자신의 블록체인에 추가하는 방식이다. 이때 2/3 검증은 참여 노드 수가 아니라 검증을 위해 네트워크에 잠근 지분의 2/3 이상이 투표되었는지 확인하는 것이다.

Tendermint는 두 번의 브로드캐스팅 과정을 통해 블록체인의 모든 노드들이 같은 블록을 합의할 수 있도록 하였다. LFT는 두 번의 브로드 캐스팅이 아닌 한 번의 브로드 캐스팅 과정을 통해 블록 데이터 검증여부를 전파하게 하였고 이 투표 결과를 다음 블록 생성자가 블록에 취합하여 넣는 방식으로 합의가 이루어진다. 이러한 방식을 통해 기존 PBFT기반 합의 알고리즘이 $2n^2 + n$ 만큼의 통신량을 가지는 반면 LFT는 최적화를 통해$n^2 + n$의 통신량을 가지도록 하였다. 그러나 Tendermint의 경우 모든 노드가 Precommit 단계에서 모든 블록이 합의되는 반면 LFT는 다음 리더가 비잔틴일 경우 몇몇 노드는 합의가 늦어질 수 있으며 다음 투표 단계에서 모든 노드의 합의가 이루어진다. LFT는 Tendermint와 같이 매 블록 생성시 리더를 변경하도록 하였으며 블록 투표가 실패하면 Round를 증가시켜 블록 생성을 시도하도록 하였다.

2.2 ScaleableBFT

ScaleableBFT는 기존 연구인 Tangaroa와 JUNO를 개선하여 많은 수의 노드 환경에서도 합의를 가능하게한 분산합의 알고리즘이다. [12][10][11]

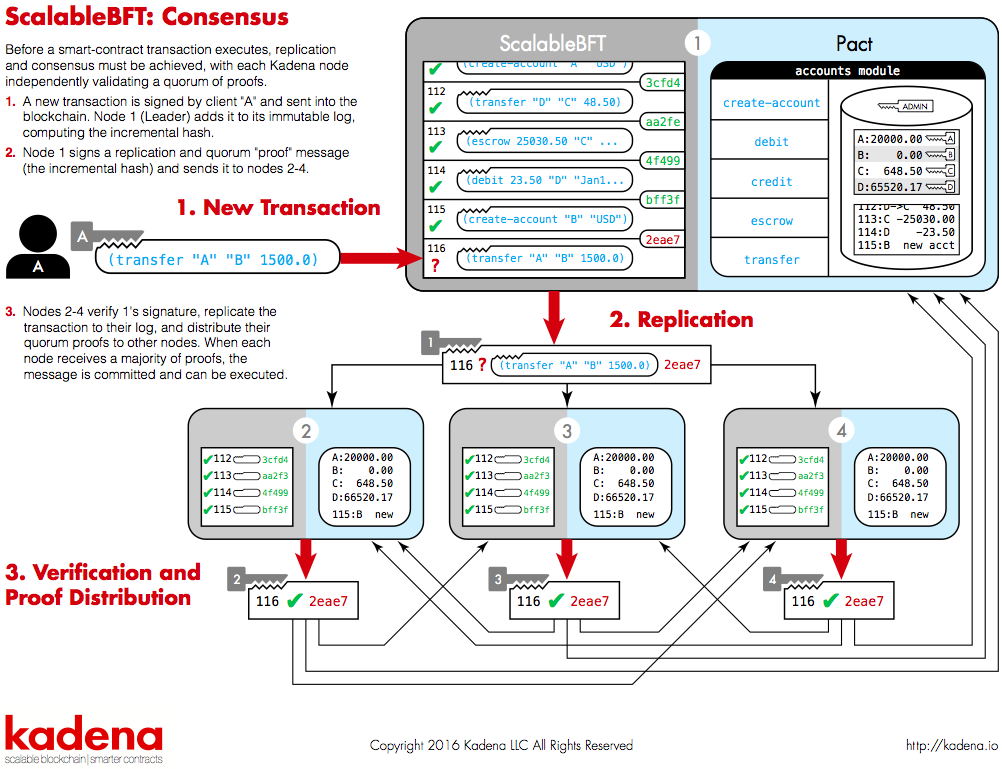

Figure 2: ScaleableBFT 합의 과정[12]

그림 2는 ScaleableBFT 동작 방식을 나타낸 그림이다. 각 리더가 트랜잭션의 순서를 정해 네트워크의 모든 노드에 트랜잭션과 증분 해시를 네트워크에 전파하면 각 노드는 자신의 트랜잭션 모음에 해당 트랜잭션을 추가시키고 증분 해시를 비교해 다른 노드들에 전파한다. 전체의 50%이상의 노드가 공유한 데이터에 대해서 자신의 트랜잭션 세트에 추가한다.[16]

ScaleableBFT는 블록체인의 전통적인 블록 구조를 벗어나 매 트랜잭션을 해시 연결이 아닌 증분 해시를 통하여 검증한다. 또한, 각 노드가 직접 정족수 이상의 노드에 해당 트랜잭션이 추가되었는지 확인하고 스스로 트랜잭션을 추가한다. 이러한 기법을 통해 블록체인을 고성능으로 바꾸었다. 또한 리더변경 시 새로운 리더가 자신의 리더 증거 데이터를 모든 노드들에 전달하는 방식으로 Tangaroa나 JUNO 등의 이전 프로젝트에서 발생하는 Runaway 문제를 해결하였다. [12]

LFT는 Tangaroa와 유사한 Raft 기반의 비잔틴 합의 알고리즘으로 ScaleableBFT와 합의 과정이 비슷하다. 그러나 ScaleableBFT가 각각의 트랜잭션을 합의하는 반면 LFT는 블록을 합의한다. 또한, ScaleableBFT가 한 번의 브로드캐스팅을 통해 모든 노드가 합의를 했다는 것을 보장하지 않지만 LFT는 합의 증거 데이터를 리더가 보내는 방식을 통해 잠깐의 합의 격차가 벌어져도 결국 합의가 완료되는 방식을 취한다. 그렇기때문에 ScaleableBFT는 합의에 빠진 노드 관리를 위해 별도의 중앙화된 기구가 필요하다. 즉, LFT는 ScaleableBFT에 비해 분권화 시스템에 더 초점을 맞췄고

ScaleableBFT는 많은 노드가 참여할 수 있는 블록체인 네트워크에 더욱 초점을 맞추었다.

또한, 리더 교체 방식에 있어 큰 차이가 있다. 기존 Raft 기반 연구들이 리더에 강한 의존성을 가지고 리더 선출을 위해 별도의 프로세스가 필요하지만 LFT는 매 블록 생성 시에 리더가 교체되기 때문에 단일 리더로 인해 발생하는 문제를 최소화하였으며 장애 리더 복구 알고리즘이 합의 과정 내에 있어 별도의 프로세스 없이 신속하게 리더 교체를 수행할 수 있다.

3 LFT 알고리즘

본 장에서는 LFT 알고리즘에 대해 설명한다. LFT 알고리즘은 블록체인을 위한 Raft 기반 합의 알고리즘으로 비잔틴 장애를 극복할 수 있는 알고리즘이다. LFT 알고리즘은 기존 상태 기계 복제 알고리즘들처럼 리더 노드와 검증 노드로 구성된다. 리더 노드는 새로운 블록을 생성하고 검증 노드는 리더가 생성한 블록을 검증하고 검증 결과를 투표하는 임무를 수행한다.

네트워크에 참여하는 각 노드는 비잔틴 노드에 의한 메시지 조작을 방지하기 위하여 공개키 기반 구조를 가진다. 모든 노드는 네트워크에 참여하는 노드의 공개키와 각 노드의 리더 순번을 알고있다.

네트워크의 모든 정상적인 노드는 리더가 보낸 예비 블록에 대하여 유효 투표와 실패 투표 단 한 번만 수행하며 비잔틴 노드는 네트워크 분기와 정지를 위해 악의적인 메시지를 횟수에 상관없이 전송할 수 있다.

LFT 알고리즘은 2단계 합의 알고리즘이다. 1번째 단계에서 리더가 블록을 만들어 다른 모든 노드에 전달하면 2번째 단계에서 각 검증 노드들은 블록은 전파하고 투표 결과를 다른 모든 노드에게 전파한다. LFT 알고리즘은 f가 비잔틴 노드의 갯수라고 할때 3f+1 이상의 노드가 존재하는 상황에서 비잔틴 장애를 극복할 수 있으며, 정족수 n-f 인 합의 알고리즘이다. 또한 각 노드는 Local Timer를 가지고 있어 리더의 통신 장애를 감지한다.

3.1장에서는 합의 알고리즘에 사용하는 블록과 투표에 대한 데이터 구조에 관해 설명하고, 3.2장에서는 알고리즘의 동작 방식에 대하여, 3.3장에서 리더 장애에 따른 리더 교체 알고리즘에 대하여, 마지막 3.4장에서는 비잔틴 노드의 공격 시나리오와 이를 LFT가 어떻게 극복할 수 있는지에 대해 설명한다.

3.1 메시지 데이터 구조

3.1 장에서는 블록체인 네트워크를 구성하는 리더 노드와 검증 노드가 전송할 수 있는 메시지에 대하여 설명한다.

그림 3은 LFT 합의 알고리즘을 사용하는 블록체인의 블록 구조이다. 다른 블록체인처럼 현재 블록의 단방향 해시값을 저장하여 해당 블록이 위변조되었는지 감지할 수 있게 하였으며 이전 블록 해시 데이터를 통해 해시 연결성을 보장한다. 그 밖에 블록 높이(블록 순번), 블록 생성 시간, 트랜잭션 머클트리와 같은 기존 블록체인에 들어가는 데이터들이 포함된다. [1][3]

구성하는 블록체인에 따라 블록체인 위에서 돌아가는 서비스의 상태를 저장하기 위한 페트리샤 머클트리 혹은 IAVL+ Tree 등의 상태 머신 저장을 위한 자료구조를 추가할 수 있다.[17][18] 본 논문은 loopchain에서 사용하는 합의 알고리즘을 일반화한

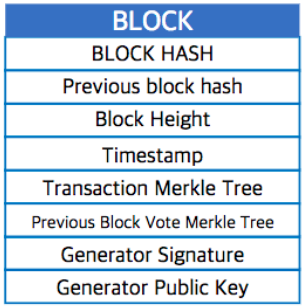

Figure 3: LFT 블록 구조

논문이기 때문에 스마트 컨트랙트 데이터를 어떤 방식으로 구성하는지에 관한 내용은 포함하지 않는다. 또한 상태 기계 복제에서 사용하는 합의 알고리즘을 차용한 합의 알고리즘을 사용하는 리더 노드의 서명을 포함한다. [3]

LFT를 구현한 블록체인의 블록은 일반적인 블록체인 네트워크에 포함되는 데이터 외에 이전 블록에 대한 투표 결과를 포함한다. Tendermint와 같은 알고리즘이 현재 블록에 대한 투표 데이터를 가지고 있는 것과는 대조적인데 이는 LFT 합의 알고리즘의 특이성에 기인한다. [3] LFT 합의 알고리즘은 각 검증 노드가 투표 데이터를 다른 모든 노드들에 전송하기 때문에 각 검증 노드는 이 단계에서 정족수 이상의 투표를 받으면 각 검증 노드가 해당 블록을 자신의 블록체인에 추가할 수 있다. 하지만 이때 네트워크의 정족수 이상의 투표를 받지 못하는 경우가 생길 수 있는데 이러한 노드의 이전 블록을 자신의 블록체인에 추가할 수 있게 하기 위해서 이후 블록에 이전 블록의 투표 데이터를 추가하여 전송한다. 또한, 허가형 블록체인의 경우 해당 블록을 검증한 사람에 대한 증거가 블록체인에 남아있는 것이 중요하기 때문에 해당 블록에 대한 투표 증거를 남기는 역할도 하게 된다.

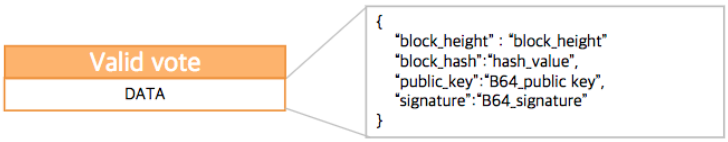

Figure 4: 블록 검증 성공 투표

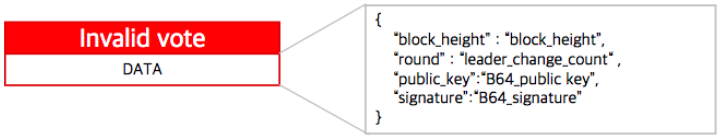

Figure 5: 블록 검증 실패 투표

그림 4와 그림 5는 이전 블록에 대한 성공과 실패 투표이다. 성공 투표에는 해당 블록을 추가할 블록의 높이와 블록의 해시 데이터가 들어가며 실패 투표에는 해당 블록이 원래 들어가야 할 블록 높이와 해당 높이에서 몇 번째 블록 생성에 실패했는지에 대한 데이터를 포함한다.

검증 성공 투표에는 블록 해시와 블록 높이를 검증 성공 투표에 포함하여 각 노드가 같은 블록에 대해 투표를 하였는지 검증할 수 있다. 그러나 검증 실패 투표에는 블록 해시에 대한 정보가 포함되어 있지 않다. 이는 시간 초과에 대한 메시지도 블록 검증 실패 투표에 포함하기 위하여 설계한 것이다. 또한 블록 검증 실패 투표에는 블록 실패 횟수(Round)를 포함하는데 이는 같은 블록 높이에서 블록 생성을 여러 번 실패할 수 있고 비잔틴 노드가 다른 노드의 실패 메시지를 반복하여 네트워크에 전파해 네트워크를 중지 시킬 수 있는 문제를 방지하기 위한 데이터이다.

3.2 알고리즘

3.2장에서는 LFT 알고리즘이 동작하는 방식에 대해 설명한다. 대체적으로 LFT 알고리즘의 합의 성공 시나리오는 Tangaroa 알고리즘과 유사하다.[10] 그러나 LFT는 리더 실패에 대한 해결 방안이 다르고 또한 투표 메시지를 충분히 받지 못한 노드를 방지하기 위한 대응 방안을 추가했다는 점이 다르다.

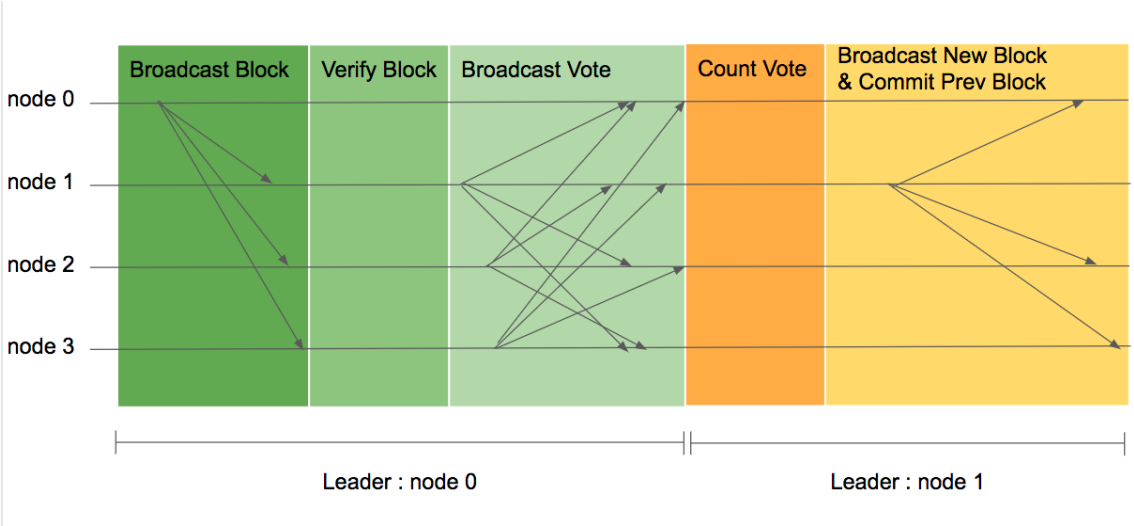

Figure 6: LFT 합의 알고리즘 성공 케이스

그림 6은 LFT 분산합의 알고리즘의 도식도이다. 각 수평선은 블록체인 네트워크에 참여하는 노드를 나타내며 화살표는 한 노드가 다른 노드에 보내는 메시지를 의미한다. 네트워크 이벤트의 순서는 왼쪽에서 오른쪽으로서 왼쪽에서 일어난 이벤트가 오른쪽에서 일어난 이벤트보다 먼저 일어난 이벤트이다. 가장 밑의 평행선은 기간에 따른 리더를 표기한다.

합의가 시작되면 검증 노드들은 리더 노드에 처리하기 원하는 트랜잭션을 전송한다. 리더 노드는 수집한 트랜잭션을 이용하여 블록을 생성하고 자신의 서명과 함께 다른 모든 검증 노드에 전송한다.(Broadcast Block) 각 검증 노드들은 블록을 받으면 1) 현 리더가 블록을 생성했는지 확인하고, 2) 블록의 높이와 이전 블록 해시가 올바른지 확인 3) 블록의 메시지가 올바른지 확인한다.(Verify Block) 검증 노드는 검증 결과에 따라 투표 메시지를 모든 노드들에게 보낸다.(Broadcast Vote)

투표 메시지를 전체 노드에 전파하는 것은 매우 중요하다. 리더 노드가 비잔틴일 경우 정족수 이상의 노드들에만 블록을 전파하여 특정 노드들을 네트워크로부터 분리하도록 시도할 수 있다. 이러한 문제를 방지하기 위해 모든 노드에 투표 메시지를 전파한다. 블록을 못 받은 노드는 블록이 생성되었는지에 대한 정보를 알 수 있고 다른 노드에 블록을 요청할 수 있다. 알고리즘 구현 시 비잔틴 노드가 생성한 가짜

투표를 제외한 확실한 블록을 요청하기 위해선 f+1 개 이상의 같은 투표를 받을 때까지 기다리다가 블록을 요청할 수 있다.

각 노드가 정족수 이상의 투표를 받게 되면 해당 높이의 블록을 자신의 블록체인에 추가할 수 있다. (Count Vote) 리더 노드는 정족수 이상의 투표를 받으면 해당 높이의 블록생성을 성공으로 취급하고 다른 노드에서 받은 트랜잭션을 모아 새로운 높이의 블록을 생성하고 다른 모든 노드에 새로운 높이의 블록 메시지를 보낸다. 생성된 블록이 제니시스 블록이 아닌 경우 앞에서 검증 노드들은 앞서 검증한 1-3외에 블록에 포함된 이전 블록 투표 검증 메시지들이 올바른지 검증한다. 같은 블록 해시에 대한 투표 메시지가 정족수 이상 포함되어 있는지 검증한다. 투표 메시지 공유 시 정족수 이상의 투표를 받지 못한 노드도 이전 블록을 자신의 블록체인에 추가할 수 있다.

블록체인은 신뢰가 부족한 노드들이 모여 데이터 분산 합의를 통해 신뢰 네트워크를 구축하는 기술이다. 허가형 블록체인 환경의 경우 각 기관이 노드로써 블록체인에 참여할 수 있다. 기존 상태 기계 복제 시스템처럼 모든 상태 기계에 응답을 보장하는 것이 아닌 각 노드가 서비스를 제공하고 트랜잭션을 생성하기 때문에 리더 노드가 블록 생성 시 특정 노드의 트랜잭션을 거부할 경우 문제가 생길 수 있다. 이러한 문제를 최소화하기 위해 Spinning 기법을 이용해 매 블록 생성 시 마다 리더를 교체하여 비잔틴 리더에 의해 발생할 수 있는 서비스 장애를 요소를 줄였다.[13]

3.3 리더 장애 극복 시나리오

3.2장에서는 LFT 알고리즘에서 리더가 비잔틴이 아니고 블록 생성이 성공한 경우에 대해서 언급하였다. 3.3장에서는 리더가 비잔틴일때 어떠한 방식으로 네트워크가 장애를 극복하고 해당 높이의 블록을 다시 생성하는지에 대해 기술한다.

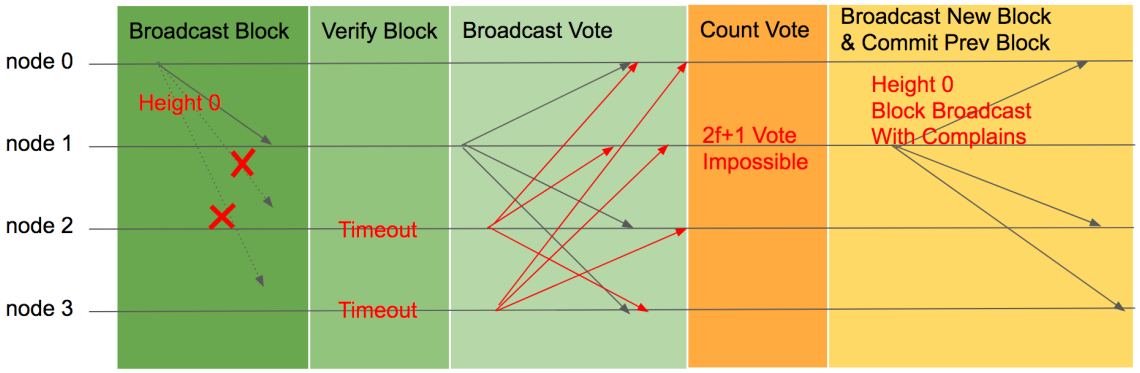

Figure 7: LFT 합의 알고리즘 실패 케이스

그림 7은 리더 노드의 장애로 모든 노드에게 블록 메시지를 전달하지 못한 사례이다. 각 노드는 리더 노드에서 전달 받은 블록에 문제가 있거나 시간내로 새로운 블록이 생성되지 못한 경우 실패 투표를 모든 노드에 전송한다. 다음 리더는 블록의 투표가 정족수를 넘지 못할 만큼 투표 실패 투표를 전해 받으면 해당 데이터를 실패한 블록 높이에 대한 블록을 다시 생성한다. 이때 이전 블록 높이의 투표와 실패 투표를 다시 넣어서 블록을 생성한다.

각 노드의 내부 타이머는 블록 검증 투표를 모든 노드에게 전송 할 때 초기화하고 새로운 리더로부터 블록을 받을 때 까지 동작한다. 이러한 타이머 동작 방식은 새로

운 리더의 연속적인 실패를 감지할 수 있도록 만들어준다. 각 노드의 내부 타이머는 정상적인 투표 상황과 시간 초과 상황의 시간 초과 세팅을 다르게 해야 한다.

LFT의 리더 교체 알고리즘은 기존 연구들과 같이 합의를 중단시키고 별도의 과정을 통해 리더를 교체하는 것이 아닌 합의 알고리즘 속에 리더 교체 알고리즘을 추가하였다. 이를 통해 알고리즘 구현을 단순화하였으며 합의 중간에 리더 교체 과정으로 진입할 때 생기는 지연 시간을 최소화하였다.

3.4 공격 시나리오 검증

본 장에서는 합의 알고리즘의 안정성을 검증하기 위해 네트워크 분기와 네트워크 중지를 위해 취할 수 있는 공격 시나리오와 각 시나리오별로 LFT가 어떻게 이러한 장애를 극복할 수 있는가에 관해 기술한다. 아래는 비잔틴 노드가 수행할 수 있는 네트워크 공격 시나리오이다.

1. 비잔틴 검증 노드에 의한 정상 블록 실패 시도
2. 비잔틴 리더에 의한 블록 분기 시도
3. 비잔틴 리더에 의한 네트워크 중지 시도
4. 비잔틴 리더에 의한 서비스 속도 저하 시도
5. 비잔틴 리더에 의한 특정 노드의 트랜잭션 거부

네트워크의 전체 노드 수가 n개이고 비잔틴 노드 수가 f개라면 블록체인 네트워크를 구성하는 전체 노드 수 n은 3f+1개 이상이어야 비잔틴 장애를 극복할 수 있다. 즉, 노드 수 n개의 네트워크에선 (n-1)/3 만큼의 비잔틴 노드의 공격을 방어할 수 있다. 알고리즘에서 허용하는 정족수는 n-f개로 3f+1 네트워크에서 2f+1개이다. 네트워크 전체 노드를 3f+1개라고 가정하자.

LFT 알고리즘은 블록의 실패를 확인하기 위해서는 f+1개 이상의 실패 투표를 받아야한다. 모든 실패 투표에는 서명이 들어가기 때문에 비잔틴 노드가 임의로 만들 수 없기 때문에 f 이하의 비잔틴 노드 환경에서는 정상적인 블록을 실패 처리할 수 없다.

공격 시나리오 2번 또한 마찬가지다. 비잔틴 리더는 네트워크를 분기 시키기 위해 같은 높이에 서로 다른 두 블록을 생성할 수 있다. 3장에서 언급 하였듯이 정상적인 노드는 같은 높이 같은 리더가 만든 블록을 단 한번만 서명을 한다.

$$
\begin{gathered}
Condition(doublecommit) = Vote_canvote <= Vote_doublecommit \\
Vote_all = 3f + 1 \\
Vote_byzantine = f \\
Vote_canvote = Vote_a + Vote_b = 4f + 1 \\
Vote_doublecommit = 2 * (Vote_all - Vote_byzantine) = 4f + 2 \\
Vote_doublecommit > Vote_canvote
\end{gathered} \tag{1}
$$

수식 (1)은 공격 시나리오 2번이 불가능하다는 것을 보여준다. 비잔틴 노드가 두 가지 블록을 같은 높이에 전부 추가시키려면 네트워크에 전파되는 투표의 개수가 두 개의 블록을 모두 검증할 수 있는 갯수 이상이 되어야 한다. 모든 비잔틴 노드 f가 동일 높이에서 이중 투표를 하여 두 개의 블록을 전부 합의에 성공하게 하려면 총 4f+2

개의 투표가 필요하다. 그러나 f개의 비잔틴 노드가 이중 투표를 하여도 만들 수 있는 투표의 개수는 최고 4f+1개 이기 때문에 두 블록을 전부 허용하는 것이 불가능하다.

공격 시나리오 3번은 로컬 타이머를 이용한 시간 초과 트리거와 리더 교체 알고리즘 때문에 불가능하다. 리더 노드는 네트워크를 중지시키기 위해 블록을 여러 개 만들어 합의 실패를 유도하던가 블록을 생성하지 않으려 할 것이다. 그러나 다음 리더는 모든 노드의 투표 메시지를 수집해 현재 높이에서 합의할 수 없다는 것을 증명하고 해당 높이의 새로운 블록을 생성한다.

공격 시나리오 4번과 5번은 합의 알고리즘 자체로는 막을 수 없으나 매 블록 생성 시 리더를 교체하는 Spinning을 통하여 피해를 최소화할 수 있다.[13] 리더가 매번 교체되기 때문에 특정 노드의 트랜잭션을 거부하는 경우도 결국은 해결될 수 있으며 리더가 매번 시간초과 시간에 맞춰 블록을 생성하려는 시도에 대한 피해 또한 최소화할 수 있다. 실제 구현에서는 블록체인 정책에 따라 이러한 노드들을 감지하여 네트워크에서 제외할 수 있다.

4 loopchain

loopchain은 LFT 알고리즘을 지원하는 최초의 블록체인이다. loopchain인은 Python3으로 구현하였으며 통신 프레임워크로는 구글 gRPC를 이용하였다.

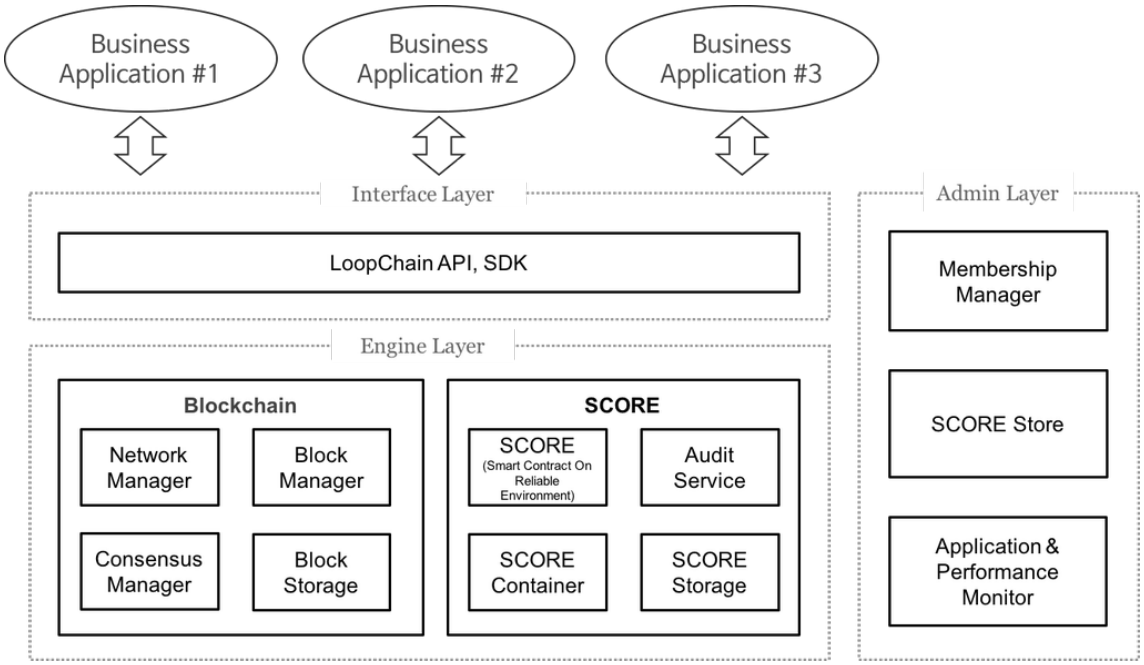

Figure 8: Loopchain 구조

그림 8은 loopchain의 모듈 구조이다. Blockchain 모듈은 노드간 합의를 위한 모듈로써 블록 데이터와 합의를 주관한다. Blockchain 모듈의 Consensus Manager 모듈에서 LFT 합의 알고리즘을 구현하였다. 서비스의 특성에 따라 다양한 다른 합의 알고리즘을 쉽게 교체하여 적용할 수 있다. loopchain은 BFT-SMaRt와 같은 선행연구처럼 낮은 지연시간과 높은 처리량을 보장하기 위해 필요한 구성요소들을 프로세스로 분리하였다.[19] SCORE(Smart Contract on Reliable Environments)는 loopchain의 스마트 컨트랙트로 구조적으로 Blockchain과 분리되어 있으며 내부 gRPC 통신을 통해 Blockchain의 요청을 받고 결과를 반환한다. 또한 블록체인 네트워크에 참여하는

노드들의 권한을 관리하기 위한 Admin Layer를 통해 노드들의 권한및 인증서를 관리하고 네트워크의 모든 노드들에게 알려 허가하지 않은 노드가 네트워크에 참여하는 것을 막았다.

5 결론

LFT는 블록체인을 위한 경량화된 고성능 합의 알고리즘이다. LFT는 비잔틴 장애를 극복하는 Raft 알고리즘의 일종으로 투표 전체 전달을 통해 비잔틴 리더에 의한 네트워크 중지 공격을 방지하였으며 블록 데이터에 투표 증거를 남겨 네트워크 장애로 투표 데이터를 충분히 받지 못한 노드도 별도의 투표 전체 전달 과정 없이 동기화할 수 있게 하였다. 또한 기존 알고리즘들이 블록에 관한 투표와 리더에 관한 투표가 분리되어 리더 변경시 네트워크에 무리를 줬던 문제를 메시지 통합을 통해 해결하였다. LFT는 분기가 일어나지 않는 즉각적인 거래 확인 및 효율적인 거래 메세지 교환이 필요한 고성능의 블록체인을 만드는데 도움을 줄 것이다.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] M. Swan, *Blockchain : Blueprint for a New Economy.* O'Reilly Media.

[3] J. Kwon, "Tendermint: Consensus without mining," 2014.

[4] C. Cachin, "Architecture of the hyperledger blockchain fabric," 2016.

[5] L. Lamport, "The implementation of reliable distributed multiprocess systems," *Computer Networks (1976)*, vol. 2, no. 2, pp. 95–114, 1978.

[6] M. Castro, B. Liskov, *et al.*, "Practical byzantine fault tolerance," in *OSDI*, vol. 99, pp. 173–186, 1999.

[7] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 4, no. 3, pp. 382–401, 1982.

[8] L. Lamport, "The part-time parliament," *ACM Transactions on Computer Systems (TOCS)*, vol. 16, no. 2, pp. 133–169, 1998.

[9] D. Ongaro and J. K. Ousterhout, "In search of an understandable consensus algorithm.,"

[10] C. Copeland and H. Zhong, "Tangaroa: a byzantine fault tolerant raft," 2016.

[11] Kadena-io, "JUNO." https://github.com/kadena-io/juno.

[12] G. S. Samman, "Kadena: The First Real Private Blockchain." https://www.linkedin.com/pulse/kadena-first-real-private-blockchain-george-samuel-samman.

[13] G. S. Veronese, M. Correia, A. N. Bessani, and L. C. Lung, "Spin one's wheels? byzantine fault tolerance with a spinning primary," in *Reliable Distributed Systems, 2009. SRDS'09. 28th IEEE International Symposium on*, pp. 135–144, IEEE, 2009.

[14] Bitshares, "Delegated Proof-of-Stake Consensus." https://bitshares.org/technology/delegated-proof-of-stake-consensus/.

[15] PeerCoin, "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake."

[16] M. Bellare, O. Goldreich, and S. Goldwasser, "Incremental cryptography: The case of hashing and signing," in *Annual International Cryptology Conference*, pp. 216–233, Springer, 1994.

[17] E. Foundation, "Ethereum Patricia-Tree." https://github.com/ethereum/wiki/wiki/Patricia-Tree.

[18] E. B. Jae Kwon, "Cosmos White Paper." https://cosmos.network/whitepaper.

[19] A. Bessani, J. Sousa, and E. E. Alchieri, "State machine replication for the masses with bft-smart," in *Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on*, pp. 355–362, IEEE, 2014.