# 자체 블록체인 네트워크 구축을 위한 플렉서블 블록체인 프레임워크 구현

하현수[O1] 정구익[1] 김명호[2] 김영종[2]
[1,2]숭실대학교 소프트웨어 학부
dhy03196@naver.com, rndlr96@gmail.com, kmh@su.ac.kr, youngjong@ssu.ac.kr

# An Implementation of Flexible Blockchain Framework(FBF) for Construct Own Blockchain Network

Hyunsoo Ha[O1] Guik Jung[1] MyungHo Kim[2] YoungJong Kim[2]
[1,2]Dept. of Software, Soongsil University

**요 약**

블록체인은 중앙 서버 없이 모든 노드가 피어(Peer)로 참여하는 퓨어 P2P 네트워크 기반의 분산 원장 기술이다. 모든 노드가 합의를 기반으로 참여하게 되어 원장의 위·변조를 사전에 방지할 수 있고, 중앙화된 서버가 존재하지 않기 때문에 탈중앙화(Decentralization)라는 특성을 가지고 있다.

이러한 특성 때문에, 다양한 서비스에 블록체인을 접목하려는 시도가 늘고 있다. 다양한 서비스들의 각 목적에 맞는 블록체인을 처음부터 설계 및 구축하는 것은 많은 시간이 소요되기 때문에 대다수 업체는 기존 프로젝트를 수정하여 자체 블록체인을 구축한다. 그렇지만 기존 블록체인의 경우 블록생성시간, 블록사이즈 등의 블록체인 속성값들이 코드 전역에 분산되어 있을뿐더러 수정을 고려하지 않고 고정되어 있어 수정 시 코드 의존성 등의 문제가 발생할 수 있기 때문에 효율적인 개발이 어렵다. 또한, P2P 네트워크의 구조를 피어에게 알려주는 역할을 하는 Seednode의 주소값이 고정되어 있어 블록체인의 속성값을 바꾸더라도 자체 네트워크로 분리시키기 어렵다.

본 논문에서는 위와 같은 문제를 해결하기 위해서 Flexible Blockchain Framework(FBF)를 제안하고 이에 대한 구현 및 개발 방법을 다룬다. FBF는 기존 블록체인 코드를 기능 단위로 모듈화하고, 블록체인 속성값들을 설정 파일 한곳으로 통합시켜 보다 효율적인 수정을 할 수 있도록 하고, Seednode의 주소값 또한 수정 가능할수 있도록 구현하여 블록체인 네트워크를 보다 쉽게 분리시킬 수 있도록 하였다.

## 1. 서 론

최근 블록체인이 뜨겁게 떠오르면서, 다양한 블록체인 서비스들이 쏟아져 나오고 있다. 하지만 업체가 제공하고자 하는 서비스와 잘 호환될 수 있는 자체 블록체인을 처음부터 설계 및 구현하는 것은 많은 시간과 노력이 필요하기 때문에 대부분의 프로젝트는 호환성, 속도 등의 다양한 문제점들을 감안하더라도 Bitcoin과 같은 기존 블록체인의 소스코드를 수정하여 사용한다.

기존 블록체인의 소스코드는 일반적으로 서비스 운영 과정에서 추가된 불필요한 기능이 소스 코드 내에 포함되어있어 전반적인 코드 분석에 어려움이 있다. 또한, 블록체인의 속성 정보인 블록체인 속성값들이 코드 전역에 분산되어 있어 속성값 수정이 비효율적이고 의존성 문제 등이 발생할 수 있다.

뿐만 아니라 Bitcoin을 포함한 대다수 블록체인은 Seednode 주소값과 다양한 인자들이 Assert 문장에 의해 고정되어 있어 수정 시, 컴파일 과정에서 에러가 발생하게 된다. 이 때문에 블록체인의 P2P 네트워크를 분리하는 것 자체에 큰 노력이 요구된다.

본 논문에서는 분석이 용이하도록 소스 코드를 기능 단위로 모듈화 및 리팩토링을 진행하고, Seednode와 블록체인 속성 등의 인자값들을 하나의 설정 파일에 통합하는 방식을 통해 유연성이 크게 개선된 Flexible Blockchain Framework(FBF)를 제안한다.

FBF는 가장 대중적으로 사용되는 일반적인 구조로 블록 헤더와 트랜잭션을 정의하여 호환성을 높였다.

FBF를 이용함으로써 사용자들은 보다 용이하게 블록체인 속성값을 수정할 수 있으며, Seednode의 주소값을 자체 블록체인에 맞는 IP 혹은 도메인으로 변경하여 손쉽게 독자적인 P2P 네트워크를 분리할 수 있다.

## 2. 관련 연구

### 2.1 Bitcoin

Bitcoin은 2008년 사토시 나카모토의 논문 “Bitcoin : A Peer to Peer Electronic Cash System”에 최초 기술됐던 블록체인 기술을 이용하여 2009년 개발된 최초의 암호화폐이다[1].

오픈 소스로 공개되어 있지만, 상수와 변수값의 수정이 제한되어 있어 수정 시 Assert에 의해 컴파일 에러가 발생하기 때문에 유연한 수정이 어렵다. 또한, 모듈화가 잘 되어있지 않아 소스 코드 분석에 시간이 오래 걸린다는 단점이 있지만 그럼에도 불구하고 가장 많은 블록체인들의 모체가 되었다.

### 2.2 Litecoin

Litecoin은 MIT의 Charlie Lee가 개발한 Bitcoin 기반의 암호화폐로 해싱함수를 SHA-256에서 Scrypt로 바꾸고, 블록사이즈와 최대 발행 수량을 늘린 블록체인 프로젝트이다. 블록체인 속성값 수정을 위해 Bitcoin의 Assert문장을 제거하여 최대 발행량, 해싱 방식, 블록사이즈, 블록 생성 시간 등의 블록체인 속성값을 비교적 플렉서블하게 수정 할 수 있게 되었다는 의의가 있다.

하지만 모듈화가 충분히 진행되지 않아 컨센서스 방식과 같은 큰 틀에서의 변화를 위해서는 기존 소스 코드를 새롭게 고치고 의존성을 검사하며 수정해야 한다는 점에서 여전히 Bitcoin 기반 방식의 한계점이 존재한다.

### 2.3 Peercoin

Peercoin은 Bitcoin을 모체로 하며 기존 Bitcoin의 합의 알고리즘인 작업증명(POW)을 지분증명(POS) 방식으로 변경한 프로젝트이다[2]. 수 많은 POS방식 블록체인들의 모체가 되고 있으며 Bitcoin 기반 프로젝트 중 최초로 가장 많은 부분에서 변화가 나타난 프로젝트이다.

합의 알고리즘을 의존성을 모두 고려하면서 Bitcoin의 코드를 수정하여 구현하였지만 모듈화가 진행되지 않아 기존 Bitcoin의 코드와 Peercoin의 새로운 코드가 난잡하게 섞여있어 블록체인 구축을 위해 사용하기에는 여전히 유연성과 모듈화가 부족하다.

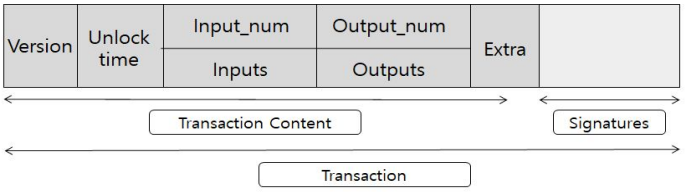
### 2.4 Bytecoin

Bytecoin은 Cryptonote 기술 기반의 최초 익명성 코인으로 Cryptonight 알고리즘을 사용하였으며 링 서명을 이용하여 익명성과 정보 보호를 강조하며 등장하였다[3]. Bitcoin 기반 블록체인과 전혀 관련이 없는 최초의 독립 블록체인이기도 하다. Cryptonote라고 하는 자체 블록체인 코어 기술을 통해 만들어졌으며, 비교적 쉽게 블록체인 속성값을 수정할 수 있지만 소스코드 길이가 Bitcoin보다 훨씬 방대하고 모듈화가 진행되지 않아 소스 분석이 아주 복잡하며, 블록체인의 속성값 변경은 간단하더라도 자체 블록체인의 P2P 네트워크를 분리해내는 것이 어렵다는 단점이 있다.

## 3. FBF(Flexible Blockchain Framework)

FBF는 기존 블록체인의 문제점을 해결하여 유연하게 블록체인 속성을 수정할 수 있도록 하며, Seednode의 주소값 설정 또한 플렉서블하게 수정 가능하도록 하여, Seednode의 주소값을 바꾸는 것으로 자체 블록체인 네트워크를 손 쉽게 분리해낼 수 있다.

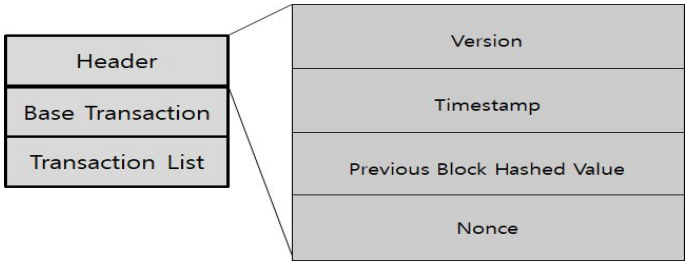
### 3.1. 트랜잭션 설계 구조




**<그림 1 트랜잭션 설계 구조>**

FBF의 트랜잭션 설계 구조는 **<그림 1>**과 같으며, 비트코인의 트랜잭션 구조를 참고하여 가장 일반적으로 사용되는 구조로 설계하여 호환성을 높였다. UnlockTime에는 UNIX의 시간정보인 Timestamp를 기록하며, Inputs에는 트랜잭션 소유권에 대한 키 정보와 Block Height가, Outputs에는 전송될 코인의 수량과 송금 주소정보가 저장된다.

### 3.2 블록 설계 구조

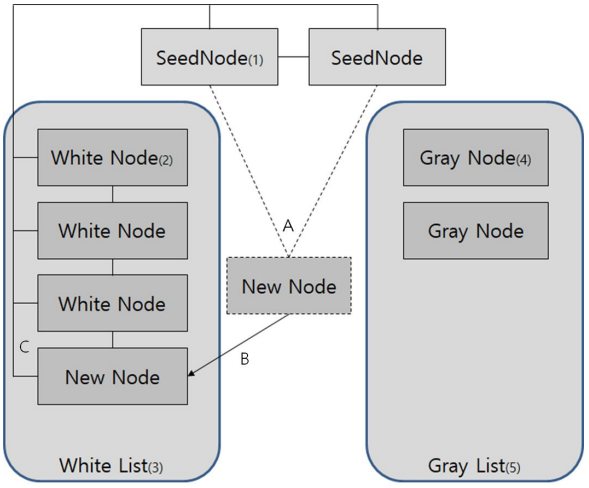



**<그림 2 블록 설계 구조>**

블록은 상기 **<그림 2>**와 같이 블록헤더, Base Transaction, Transaction List로 구성하였다.

블록헤더는 Version과 UNIX기반의 Timestamp,이전 블록의 해시값, 작업증명을 위한 Nonce로 구성되어 있으며 이러한 값들을 통해 위·변조를 파악할 수 있다. Base Transaction은 해당 블록에 생성된 최초 트랜잭션인 블록 생성 트랜잭션 정보를 저장한다. 이를 통해 채굴 보상을 받을 사람을 식별할 수 있다. Transaction List에는 해당 블록에 포함된 트랜잭션들의 TxHash값이 리스트 형태로 저장된다.

### 3.3 P2P 네트워크 설계 구조 및 구현




**<그림 3 P2P 네트워크 설계 구조 및 구현>**

**<그림 3>**은 FBF의 P2P 네트워크 개발 구조이다.

**(1)** Seednode는 네트워크의 첫 번째 노드이며 다른 노드들은 처음 연결 시 해당 노드에 먼저 연결되어 현재 네트워크에 참여하고 있는 노드들의 항목을 받아 항목의 노드들과 연결된다.

**(2)** White Node는 현재 네트워크에 P2P연결되어 현재 동작 중인 노드를 의미한다.

**(3)** White List는 White Node들의 집합이다. 이는 새로운 노드가 연결될 때 Seednode에서 전달되는 정보이다.

**(4)** Gray Node는 블록체인 네트워크에 참여하였지만, 현재는 연결이 종료된 노드를 의미하며 각 Gray Node는 종료 당시의 White List를 갖고 있다.

**(5)** Gray List는 연결이 종료된 노드인 Gray Node들의 집합이다.

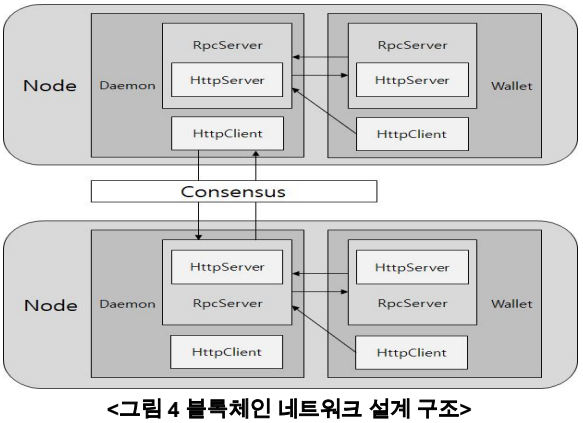
**동작 A**는 새로운 노드가 블록체인 네트워크에 참여하고자 할 때 SeedNode와 연결되어 현재 참여자 노드 항목인 White List를 받아오는 동작이다.

**동작 B**는 White List에 새로운 노드가 추가되는 동작이다.

**동작 C**는 White List에 있는 모든 노드와 연결되는 동작이다.

위와 같은 설계를 통해, Seednode의 변경만을 통해 손쉽게 분리된 P2P 네트워크를 구성할 수 있다.

### 3.4 블록체인 네트워크 설계 구조




**<그림 4 블록체인 네트워크 설계 구조>**

**<그림 4>**는 FBF의 P2P노드의 구조와 통신 방식 모델이다.

P2P 네트워크를 구성하는 각 노드는 데몬과 지갑을 가지고 있으며 이들은 RPC 서버와 HTTP 서버를 사용하여 외부 노드 및 프로세스와 통신한다. 이러한 통신 방식을 통해 동기화 과정 및 합의를 통한 블록 생성 등의 과정을 진행하게 된다.

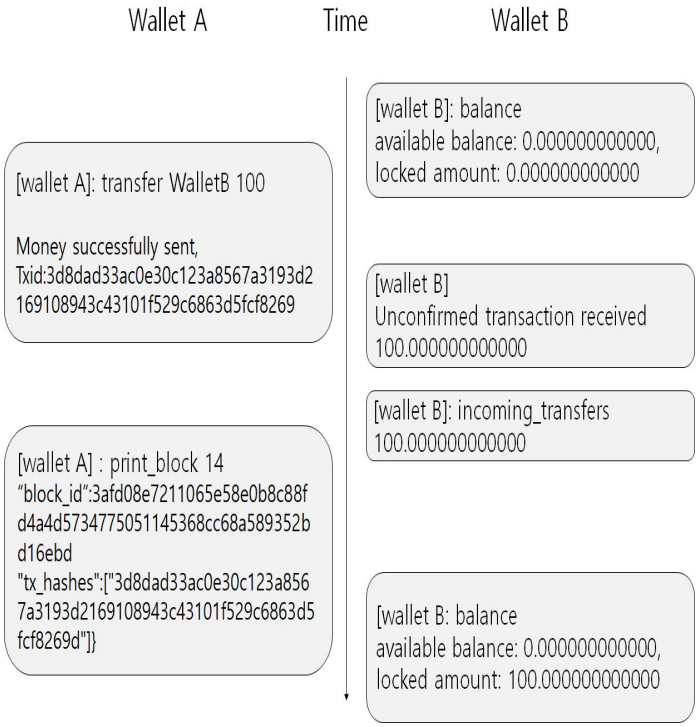
RPC 메소드는 Bytecoin의 JSON_API 오픈 소스를 이용하여 구현하였다[4][5]. 노드 및 프로세스 간 통신 시, 요청 측에서 JSON 방식으로 작성된 시그널과 메소드 실행 요청을 전달하고 그에 따른 State값 또는 JSON 결과를 반환하는 방식으로 구현되었다.

FBF는 합의 방식으로 작업증명(POW)을 사용한다. 작업증명 방식이란 컴퓨터의 연산력을 바탕으로 합의를 진행하는 방식으로 새로운 블록을 블록체인 네트워크에 추가하기 위해서는 채굴 난이도에 따른 특정 규격을 만족시키는 nonce값과 그 해시값을 구해야 한다. 본 프레임워크에서는 임의의 nonce값을 해시함수에 넣고 채굴 난이도에 따른 숫자보다 작은 값인지 확인한 후 만약 채굴 난이도에 조건에 맞는 nonce를 구하게 된다면 블록을 생성한 뒤 채굴 보상을 받고, 아니라면 또다른 임의의 nonce값을 대입하며 조건을 만족하는 블록 해시 값을 찾도록 구현되었다.

### 3.5 자체 블록체인 구현 및 분리 실험

본 논문에서 개발한 FBF를 사용하여 구현된 자체 블록체인의 Seednode 주소를 Google Cloud Instance의 IP주소로 대체한 뒤, 블록체인의 가장 기본적인 기능인 송금 기능을 실험하여 정상적으로 블록체인이 구현되었는지를 확인한다.

#### 3.5.1 송금 트랜잭션 실험 과정




**<그림 5 송금 트랜잭션 실험 과정>**

**<그림 5>**는 FBF를 이용하여 구현된 자체 블록체인의 지갑 A에서 지갑 B로 송금하는 과정을 나타낸다. 각각의 지갑은 서로 다른 두대의 PC에서 동작하고 있는 상태로 실험하였다.

지갑 B의 터미널에서 balance 명령어를 통해 지갑에 존재하는 코인을 확인 후 지갑 A의 터미널에서 transfer 명령어를 통해 지갑 B로 100개의 코인을 전송한다. 이후 트랜잭션이 위치한 블록인 14번째 블록의 생성이 완료된 다음 지갑 B의 터미널에서 다시 balance 명령을 호출한 결과 자체 코인 100개가 성공적으로 지갑 A에서 지갑 B로 전송된 것을 확인할 수 있다.

#### 3.5.2 송금 트랜잭션 실험 결과

```
'jsonrpc' : '2.0',
  'id' : 'transfer' ,
  'result' : {
      'transaction' : {
        'fee' : 10,
        'extra' : '0178a684d7d2e802c3a15854ab011bd78...',
        'timestamp' : 0,
        'blockIndex' : 13,
        'state' : 0,
        'transactionHash' : '3d8dad33ac0...',
        'unlockTime' : 0,
        'transfers' :
{
        'amount' : 100,
        'type' : 0,
        'address' : 'FUACdVN272wipNYR..."
}
        'paymentId' :,
        'isBase' : False
      }
```

**<그림 6 트랜잭션 결과>**

**<그림 6>**은 실제 트랜잭션의 결과로 나온 JSON 값이다. amount값이 송금을 요청한 코인의 개수인 100임을 확인할 수 있으며, 송금 목적지인 지갑 B의 주소를 address에서 확인할 수 있다. 이를 통해 자체 블록체인에서 성공적으로 송금이 진행된 것을 확인할 수 있다.

본 과정은 블록체인의 가장 기본적인 기능인 송금을 실험한 결과로, 복잡한 과정 없이 설정 파일에서 블록체인 속성값과 Seednode의 주소값을 변경한 것만으로 블록체인 네트워크가 독립적으로 분리되었다는 것을 확인 할 수 있다.

## 4. 결론 및 향후 연구

본 논문에서는 기존 프로젝트에선 곳곳에 흩어져있던 블록 생성 속도, 블록 사이즈 등의 블록체인 속성값들을 하나의 설정 파일로 통합하여 속성값 변경이 보다 간편하며, Seednode를 통한 P2P 네트워크 접속 방식을 통해 단순히 Seednode의 IP를 수정 하는 것 만으로도 간편하게 블록체인 네트워크 분리가 가능하도록 하는 플렉서블 블록체인 프레임워크(FBF)를 개발 및 구현하였으며 이를 통해 독립적인 블록체인 네트워크를 손쉽게 구현할 수 있다는 것을 블록체인의 가장 기본적인 기능인 코인 송금 실험을 통해 검증하였다.

향후 연구 방향으로는 본 프레임워크의 성능을 향상시키고 보다 간단하게 수정할 수 있도록 코드를 리팩토링하여 더욱 뛰어난 확장성과 범용성을 가지는 오픈 소스로서의 블록체인 코어 프레임워크를 구현하는 것을 목표로 한다.

## 5. 참고 문헌


[1] Satoshi Nakamoto, “Bitcoin A Peer-to-Peer Electronic Cash System,” bitcoin.org, 2009

[2] Sunny King, “PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake,” semanticscholar.org, 2012

[3] Nicolas van Saberhagen. “CryptoNote v 2.0” (White Paper),” cryptonote.org, 2013

[4] Bytecoin. “Bytecoin RPC Wallet JSON RPC API,” https://wiki.bytecoin.org/wiki/Bytecoin_RPC_Wallet_JSON_RPC_API, 2012

[5] Bytecoin. “Daemon JSON RPC API,” https://wiki.bytecoin.org/wiki/Daemon_JSON_RPC_API, 2012