Informe Proyecto 1: Prototipo de Blockchain Simplificada

Integrantes

David Rodriguez, 14-10930 Gregory Muñoz, 16-11313

Diseño de la aplicacion

Estructuras principales

El diseño consta de 7 Estructuras Principales:

1. Nodo BlockChain (Aplicacion)
2. Generador de Identidades
3. Generador de Transacciones
4. Explorador de Bloque
5. Explorador de Transaccion
6. Cliente
7. Interfaz Grafica

Estructuras Internas

Ademas de contar con *Estructuras Internas* utiles para desarrollar las estructuras principales

- NodoP2P
- Wallet
- Entrada
- Gasto
- Transaction
- Block
- BlockChain
- Logger

Diseño de las estructuras internas

NodoP2P

Esta Estructura debe tener dos componentes principales:

- Cliente Socket:
    - Debe tener un Cliente socket donde guarde en un arreglo de nodos todas la conexiones con los demas nodos de la red que esten conectados con el nodo.
    - Cada conexcion se debe implementar como un hilo virtual, para mantener la conexion
    - La lista de conexiones debe estar separada en "inbound" y "outbound" haciendo referencia a cuales son los nodos que estan conectados a mi y cuales son los nodos con los que estoy conectado, respectivamente, por ejemplo:
        - Para el caso de inbound: Nodo x -> NodoPrincipal
        - Para el caso de outbound: NodoPrincipa -> Nodo x
    - Se debe tener una memoria compartida con el hilo principal de la aplicacion para saber que informacion enviar a cada nodo conectado.
- Servidor Socket:
    - Se debe tener un Servidor Socket que siempre este a la escucha de los mensajes que envian los nodos conectados al nodo en cuestion.
    - Se debe tener un manejador de eventos "Event Handler" Para saber como manejar en mensaje recibido
    - Se debe tener una memoria compartida con el hilo principal de la aplicacion para procesar el contenido del mensaje recibido.

Wallet

- Wallet:
    - Esta estructura es principalmente para generar y almacenar claves asimetricas (publicas y privadas) y un address
        - Para crear las claves asimetricas se usa GPG.
        - Para crear el address, se le aplica a la clave publica SHA1, seguido de aplicarle Encode Base58.
    - Gestionar Seguridad de mensajes entre Nodos de la red:
        - Firmar mensajes
        - Encryptar mensajes
        - Desencryptar mensajes
    - Almacenar Claves publicas y address de los nodos conectados

Entrada

- Entrada:
    - Representa una Entrada que puede a tener una transaccion
    - Almacena:
        - Una referencia al hash donde se encuentra la UTXO asociada
        - El indice de la UTXO dentro de la lista de Gatos de la transaccion referenciada anteriormente
        - El address de la wallet desde la que se esta enviando

        - La cantidad o valor de lo que se esta enviando

Gasto

- Gasto:
    - Representa un Gasto que puede a tener un transaccion
    - Almacena:
        - El indice de la UTXO dentro de la lista de Gatos de la transaccion actual
        - El address de la wallet a la que se esta enviando
        - La cantidad o valor de lo que se esta recibiendo

Transaccion

- Transaccion:
    - Representacion una transaccion que va estar en un bloque
    - Almacena:
        - Un timestamp, tiempo en el que fue creada la transaccion
        - Una lista de Entrada
        - Una lista de Gasto
        - Indice del bloque donde va a estar includa la Transaccion
        - Valor total de las Entradas
        - Valor total de los Gastos
        - Hash de la Transaccion
        - Flag para saber si la transferencia es de tipo Coinbase
        - Tamaño de la transaccion en bytes
    - El Hash de la transaccion se calcula concatenando el timestamp, el indice del bloque, y la lista de entradas y gastos y hasheando con el metodo SHA256
    - El calculo del tamaño de la transaccion es los bytes que ocupe toda la transaccion
    - Se debe crear una transaccion especial para el caso de una Transaccion de tipo CoinBase

Bloque

- Bloque:
    - Representacion de un Bloque de la Blockchain
    - Alamacena:
        - Hash del bloque anterior
        - Timestamp: tiempo en el que fue creado,
        - Dificultad del PoW para el bloque
        - Raiz del Merkle Tree
        - Indice del bloque dentro de la lista,
        - Lista de transacciones a contener
        - Hash del bloque
    - Para calcular el hash se usa SHA256 sobre la concatenacion del hash del bloque anterior, el timestamp, la dificultad y la raiz del merkle root, siempre y cuando este hash cumpla con la condicion del PoW
    - Un metodo para verificar si un bloque esta correcto
    - El merkle tree se debe generar en base a una lista de hashes de las transacciones seleccionadas para estar en el bloque

BlockChain

- BlockChain:
    - Estructura para almacenar y hacer operaciones sobre los bloques
    - Almacena:
        - La dificultad que va a tener el nodo para poder minar un bloque
        - Recompensa que obtendra el nodo que mine el siguiente bloque
        - Lista de los Bloques ya minados
    - Debe tener un metodo para generar el bloque genesis, este bloque solo debe contener la primera
    - Debe tener un metodo para Crear un Bloque
    - Debe tener un metodo para poder "Minar" el Bloque
        - Se utilizara el algoritmo de Proof Of Work (Se explicara mas adelante)
    - Debe tener un metodo para cambiar el estatus del bloque y transacciones
    - Debe tener un metodo para validar bloque
        - Se debe recrear el hash con el nonce dado y comparar con el hash del bloque ademas de cumplir con la condicion de PoW
    - Debe tener un metodo para validar transaccion
        - Validar montos
            - Que hayan entradas y gastos
            - Que la suma de las entradas sean mayor al de los gastos
        - se debe verificar que cada entrada corresponde a un gasto de una transaccion ya verificada
        - Manejar el caso especial de verificacion de una transaccion de Coinbase
    - Debe tener un metodo para insertar bloques a la lista
    - Debe tener un metodo para buscar bloque por hash y otro meotodo para buscar por indice
    - Dene tener un metodo para buscar transaccion por hash

Logger

- Logger:
    - Estrucutra para almacenar los logs que se pueda generar en un Nodo
    - Alamcena:
        - ID del Nodo
        - Lista de Logs
        - Directorio donde se almacenara un archivo con los logs
    - Debe tener un metodo para generar un log de tipo Info, debe ser unicamente informativo
    - Debe tener un metodo para generar un log de tipo error

    - Debe tener un metodo para generar un log de tipo warning
    - Debe tener un metodo para generar el archivo con todos los logs almacenados

Proof of Work

- Metodo usado como algoritmo de consenso para la generacion de un bloque
    - Como prueba de trabajo
        - Se genera el hash del bloque con cierto parametro NONCE
        - Se verifica si empieza con cierta cantidad de Ceros, que seria la dificultad
        - Si cumple, entonces la prueba de trabajo es exitosa
        - si no se cumple, se tiene que calcular un NONCE aleatorio entre 0 y 100000000000
        - se repite el proceso
    - Se esta tomando la dificultad como un entero que representa cuantos ceros tiene que tener el hash del bloque

---

Nodo BlockChain (Aplicacion)

- Nodo Blockchain hereda NodoP2P
    - Estructura principal de la red, es la responsable de generar y verificar bloques ademas de ingresar transacciones para poder ser incluidos en un bloque
    - Cada Nodo contiene una Wallet
    - Cada Nodo contiene un Logger
    - Cada Nodo contiene un BlockChain
    - Almacena:
        - Configuraciones iniciales obtenidas de un archivo YAML
        - lista de transacciones en espera
        - Tabla de hash de nodos de la red con key = nombre del nodo, value = (host, port, pubKey, lista de nodos con los que esta conectado)
    - Debe tener un metodo para empezar el proceso de minado
        - Este sera un Loop infinito (o hasta que ocurre un erro en la generacion del bloque)
        - Debe usar el metodo de generar bloque de BlockChain
        - Debe usar el metodo de minar de BlockChain
        - Si el bloque se mino exitosamente debe propagarlo por la red para que sea verificado por los demas nodos
    - Debe tener un metodo para recibir mensajes y poder manejar cada uno, ademas de responder con un mensaje de ACK
    - Debe tener un metodo para manejar el caso de quere ingresar una transaccion
        - Se debe usar el metodo de validar transaccion de BlockChain
        - Se debe validar que la transaccion no se encuentre en la lista de transacciones pendientes
        - Se debe agregar la transaccion validada a la lista de transacciones pendientes
    - Debe tener un metodo para manejar el caso de presentacion de un nodo con otro
        - Debe retornar toda la data del Blockchain y lista de transacciones pendientes para actualizar el nuevo nodo
    - Debe tener un metodo para manejar el caso de propagacion de una transaccion
        - Se debe usar el metodo de validar transaccion de BlockChain
        - Se debe validar que la transaccion no se encuentre en la lista de transacciones pendientes
        - Se debe agregar la transaccion validada a la lista de transacciones pendientes
    - Debe tener un metodo para manejar el caso de propagacion de un bloque candidato
        - Se debe verifcar si el bloque existe ya en el
        - Se debe usar el metodo de verificar bloque de BlockChain
        - Se debe usar el metodo de Agregar bloque de BlockChain
    - Se debe tener un metodo que haga el proceso de apagar el nodo
        - Usar el metodo de generar archivo de logs de Logger
        - Usar el metodo de stop de NodoP2P
        - Para el proceso de minado

Generador de Identidades

- Identities
    - Estructura para crear identidades y nodos dentro de una red, nota: esta estructura es de pruebas
    - Se inicializa con el numero de identidades y numero de nodos
    - Almacena:
        - Tabla de Hash de identidades con key = address y value = (lista de utxo, public key)
            - Cada identidad contiene una estructura de Wallet
        - Lista de Nodos de Blockchain

Generador de Transacciones

- TransactionGenerator implementa un Nodo P2P:
    - Estrucutra que interactura con las identidades y nodos de la red, nota: esta estructura es de prueba
    - Almacena:
        - frecuencia de envio de transacciones
        - Numero de Entradas minimo dentro de la transaccion
        - Numero de Entradas maximo dentro de la transaccion
        - Numero de Gasto minimo dentro de la transaccion
        - Numero de Gasto min dentro de la transaccion
        - Tabla de hash de nodos de la red con key = nombre del nodo, value = (host, port, pubKey, lista de nodos con los que esta conectado)
        - Direccion del log para almacenar los logs generados
    - Debe tener un metodo para empezar a generar transacciones cada rango de tiempo
    - Debe tener un metodo para introducir las Transacciones a la red, se puede enviar a cualquier nodo de la red

Explorador de Bloque

- BlockExplorer hereda NodoP2P
    - Estrucutra para buscar bloques en la red
    - Almacena:

        - Tabla de hash de nodos de la red con key = nombre del nodo, value = (host, port, pubKey, lista de nodos con los que esta conectado)
    - Debe tener un metodo para obtener un bloque de la red dado la altura
        - Se debe enviar un mensaje de tipo explorador a la red
    - Debe tener un metodo para obtener un bloque de la red dado un hash
        - Se debe enviar un mensaje de tipo explorador a la red

Explorador de Transaccion

- TransacExplorer hereda NodoP2P
    - Estrucutra para buscar transacciones en la red
    - Almacena:
        - Tabla de hash de nodos de la red con key = nombre del nodo, value = (host, port, pubKey, lista de nodos con los que esta conectado)
    - Debe tener un metodo para obtener una transaccion de la red dado un hash
        - Se debe enviar un mensaje de tipo explorador a la red

Cliente

- Client hereda NodoP2P
    - Estructura para manejar las UTXO de cada cliente
    - Cada cliente contiene una Wallet
    - Almacena:
        - Lista de UTXO asociadas al addres del cliente
        - Balance total del cliente
        - Tabla de hash de nodos de la red con key = nombre del nodo, value = (host, port, pubKey, lista de nodos con los que esta conectado)
    - Debe tener un metodo para generar una transaccion
        - Debe generar las entradas de la transaccion con los UTXO del cliente
        - Debe generar los gastos con las address
        - Debe verificar que la transaccion esta bien armada antes de enviarla
    - Debe tener un metodo para hacer refresh del balance del cliente
    - Debe tener un metodo para introducir las Transacciones a la red, se puede enviar a cualquier nodo de la red

Interfaz Grafica

- Interfaz Grafica
    - Interfaz que agarra los logs generados por un Nodo Blockchain y recrea el proceso de envio de transaccion y minado de bloques.

Consideraciones de la implementacion

Status del proyecto

De acuerdo al diseño propuesto, se implemento completo El Nodo Blockchain, El Explorador de Bloques y el Explorador de Transacciones, Se implemento casi completo El generador de Identidades, el generador de Transacciones y el Cliente, por ultimo, no se implemento la interfaz grafica, exceptuando la generacion del Archvio log de cada Nodo.

Un punto importante es que los mensajes no estan siendo encriptados o firmados segun sea el caso, si se generan las claves asimetris y el address segun el diseño, pero no se logro implementar esa proteccion de la data

Implementacion

Para el caso del Nodo, se uso una libreria para crear aplicaciones P2P llamada p2pnetwork, funciona de acuerdo a como se diseñó, se ejecuta el servidor y el las conexiones en hilos distintos, para el caso del proceso de minar si se hace en el proceso principal. Los mensajes entre nodos estan totalmente implementados, sin embargo, a la hora de minar nodos, los resultados fueron que un proceso siempre va a minar mas rapido que otro por lo que puede darse el caso de que un nodo solo mine y los demas solo escuchen y validen.

Para el caso de los generadores, los implementamos como si fuesen otro nodo de la red, solo que mas simples, es decir, solo generar y envian, no hay que procesar los mensajes recibidor mas alla de un mensaje de ACK

Para el caso de los exploradores, nuevamente, los implementamos como si fuesen otros nodos de red, de igual manera, solo para hacer consultas a la blockchain de un nodo

Para el caso del cliente, es una combinacion de un generador con un explorador, para poder generar transacciones pero a la vez poder revisar las UTXO asociadas al address del cliente.