Trazabilidad de tiempo UTC y Time Stamp en Blockchain

Guillermo Pérez Alba

ETSIIT - Universidad de Granada

Contexto histórico

17 de julio de 2018

1. Contexto histórico

1.1. Estudios de la Blockchain de Bitcoin

El estudio realizado en el paper titulado InformationPropagationBTCNetwork se centra en un primer momento en el retardo de la red. Se denomina retardo D (delay) al tiempo que tarda un bloque emitido por un minero en llegar al resto de nodos de la red. El resultado obtenido tras observar 10,000 bloques se observa en la figura 1. El retardo medio es de 12.6 segundos. Pasados 40 segundos aún hay un 5 % de los nodos que no han recibido el bloque.

Pese a que también se analiza el retardo en relación con el tamaño del bloque, se observa que a partir de 20 kB el retardo es constante.
El objetivo de medir este retardo es ver su interacción real con el número de forks que se suceden en la red. Cuando el primer nodo de la red consigue minar el bloque N+1 inmediatamente lo emite a la red. Desde esta emisión transcurre un tiempo hasta que la red recibe el bloque. Previamente se definió este tiempo como retardo D. Durante este tiempo D, el resto de los nodos de la red siguen minando el bloque N+1 ya que aún no se han enterado de que un nodo ya ha conseguido minar el bloque N+1. Si dentro del tiempo D algún nodo mina el bloque N+1, se tendrían 2 bloques N+1 emitidos a la red denominados (N+1) y (N+1)'. En este momento se podría dar el caso de que una parte de la red tome la cadena que acaba con el bloque (N+1) y otra parte la cadena que acaba con el bloque (N+1)'. Habría que esperar al minado de varios bloques

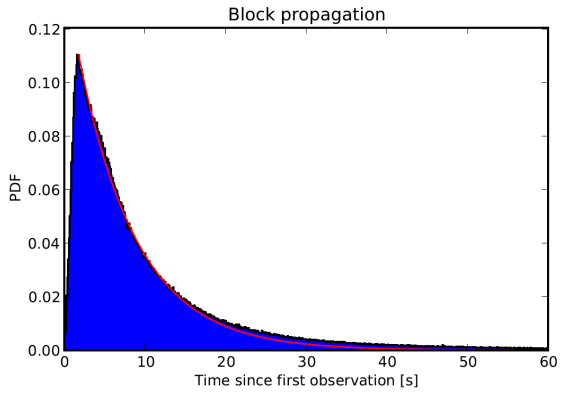

Figura 1: Block Propagation

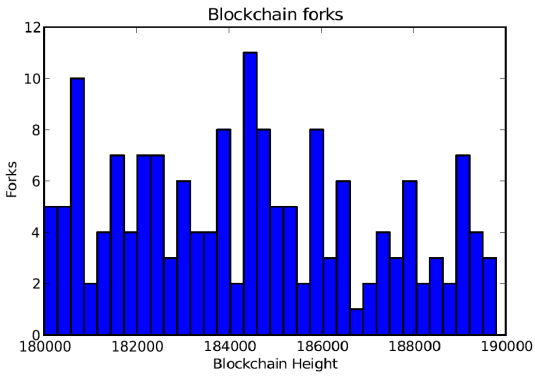

Figura 2: Blockchain forks

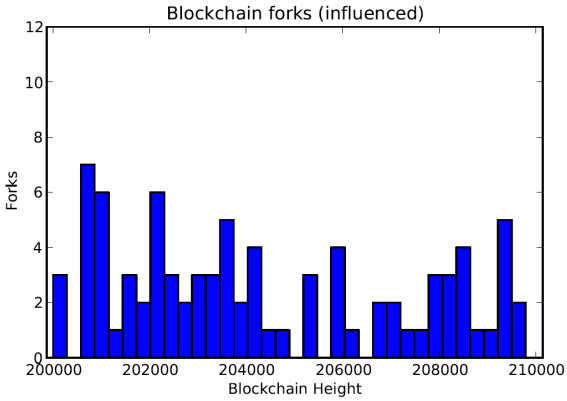

Figura 3: Blockchain forks influenced

más para que la red se ponga de acuerdo en una única cadena, suponiendo esto un desperdicio de poder computacional. Cuanto mayor sea D mayor es la probabilidad de que se produzcan diversos forks en la red. El estudio pasa a medir el número de forks que observa en un intervalo de 10,000 bloques mientras se conectan a un gran número de nodos (bastante más grande de lo normal con el objetivo de observar la mayor parte de la red posible). Se observa en la figura 2 que se producen un total de 169 forks en los 10,000 bloques (r=1.69 %).

En un intento por demostrar la repercusión de D en los forks producidos en la red, realizan un experimento en el que aumentan la conectividad de la red. Para ello montan un nodo con un total de 4000 conexiones a otros nodos (en Bitcoin lo habitual son 8 conexiones). Durante este tiempo, de nuevo miden el número de forks que se producen en la red. Los resultados mostrados en la figura 3, muestran que la tasa de forks pasa de un 1.69 % a un 0.78 %. Supone una mejora de un 53.41 % en el número de forks.

¿El número de forks sólo está relacionado con el retardo de la red D? La respuesta es NO. El tiempo de bloque (también llamado tiempo de Proof of Work) tiene una influencia similar en el número de forks. Si se disminuye el tiempo de bloque (se facilita la Proof of Work), se produce un aumento en el número de forks. Por lo tanto, si se quiere una blockchain escalable hay que tener esto en cuenta.
¿El aumento de forks sólo conlleva un desperdicio de computo? De nuevo la respuesta es NO. El paper titulado GHOST explica porqué disminuir el tiempo de bloque hace a la blockchain más vulnerable a sufrir un ataque del 51 %. La figura 5 muestra una situación en la que un atacante (una minning pool) intenta un ataque del 51 %. Se supone que el atacante tiene un poder computacional del 30 % y el resto de la red honesta dispone el 70 % restante.
Disminuir el tiempo de bloque conlleva un mayor número de forks. Cuando se producen varios forks, la parte

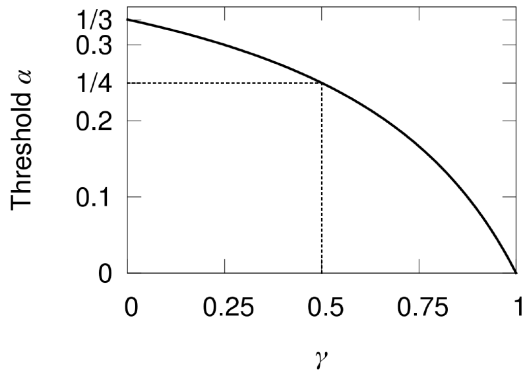

Figura 4: Relación aceptación y potencia de cómputo

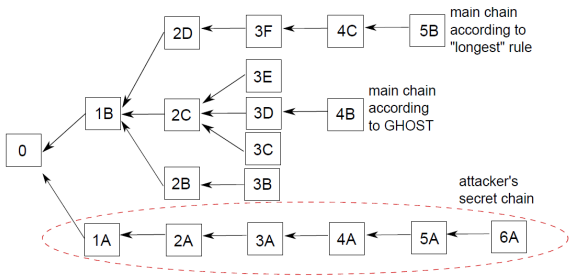

Figura 5: Algoritmo GHOST

honesta de la red está dividida en tantas partes como forks haya. En el instante 2 de la figura 4 se tienen 3 forks, por tanto, el 70 % del poder computacional de la red honesta estará dividido en estos 3 forks. En cambio, el atacante está actuando por su cuenta así que emplea el 30 % de su poder computacional en su única cadena. Es evidente que el atacante con sólo el 30 % del poder computacional de la red será capaz de crear una cadena más larga que el resto de la red honesta y llevará su ataque con éxito. El protocolo que debería seguir la minning pool que realiza el ataque se denomina Self-Mine-Strategy y aparece formalmente descrito en el paper titulado "Self-Mine-Strategy". Se obtiene teórica y experimentalmente que el límite de potencia de cómputo de un atacante, a partir del cual el algoritmo Self-Mine-Strategy triunfa, está relacionado con el ratio de nodos honestos que aceptarán la cadena del atacante como principal para minar sobre esta. La figura 4 muestra dicha relación.

Esta situación se produce en el caso de que el protocolo se base en elegir como cadena principal a la cadena más larga. Se propone en el paper un algoritmo alternativo a "la cadena más larga" que se podría usar en caso de que se reduzca el tiempo de bloque. El algoritmo propuesto permitiría que el ataque del 51 % sólo se pueda realizar con éxito en caso de que el atacante realmente tenga ese 51 % y no con un tanto por ciento menor como se ha visto antes. El algoritmo se denomina GHOST. Este algoritmo en lugar de quedarse con la cadena más larga, elige como principal al subtree más largo. Es decir, GHOST elige la cadena que acaba en 4B como cadena principal ya que tiene en cuenta todos los bloques que hay en su subtree. Los bloques 3E y 3B se tienen en cuenta en este caso y son denominados uncles.
Estos bloques se han minado durante el retardo D y aunque no se hayan minado los primeros (o no se hayan transmitido los más rápidos por la red), conllevan un computo similar al resto de bloques. Por ello, las transacciones de los uncles no se tienen en cuenta, pero si se tienen en cuenta para decidir qué subtree tiene más cómputo y se convierte en la principal.

1.2. Estudios de la Blockchain de Ethereum

Todos los estudios expuestos se han realizado sobre la red de Bitcoin. Sin embargo, dicha red es más reacia a cambios que otras como Ethereum. La red de Ethereum se construye con la intención de tener un tamaño igual o superior a la red de Bitcoin. Así que la red que realmente se ha aprovechado de todos los estudios realizados ha sido la de Ethereum, ya que plantea su protocolo en base a los resultados obtenidos en los estudios.
Una primera aproximación al funcionamiento de la red Ethereum se presenta en el White Paper. De manera más detallada se discuten las decisiones tomadas en los documentos DesignRationaleETH y Toward12secBlockTime. Todo esto lleva a un documento formal, Yellow Paper.

- A raíz del estudio del retardo de la red Bitcoin, se decide que el tiempo de bloque de Ethereum sea de 12 segundos.

- La dificultad de la PoW se calcula en base a la dificultad del bloque inmediatamente anterior y a la diferencia de timestamps entre bloques. La ecuación es la siguiente:

```
diff(genesis) = 2^32

diff(block) = diff.block.parent + floor(diff.block.parent / 1024) *
    1 if block.timestamp - block.parent.timestamp < 9 else
    -1 if block.timestamp - block.parent.timestamp >= 9
```

Figura 6: Cálculo dificultad Ethereum

- Como consecuencia del paper GHOST, en Ethereum se decide utilizar una aproximación de este algoritmo. Se elegirá el subtree con más cómputo teniendo en cuenta que se pueden incluir uncles de hasta 7 bloques atrás.

- Se establecen una recompensa de 7/8 para los bloques uncles. Dichas recompensas no son tan cuantiosas como las del bloque de la cadena más larga, pero favorecen que las recompensas no se centralicen. Los bloques que incluyan bloques uncles recibirán una recompensa de 1/32 por ser bloques nephew además de su recompensa por ser bloques de la cadena principal.

Tanto en la red de Bitcoin como en la de Ethereum el retardo de la red es uno de los principales inconvenientes para la seguridad y escalabilidad de la blockchain. En ambos protocolos se asume que ese retardo de la red no se puede mejorar, y plantean sus algoritmos en base a este. En ningún momento se intenta mejorar este retardo. Atajar el problema del retardo de la red de raíz supondría una mejora para muchos de los problemas previamente tratados. No sólo en el caso de blockchains con PoW sería una mejora, también en el caso de PoS mejoraría muchos aspectos de la red.

1.2.1. Proof of Stake

1.2.2. Sharding

1.3. Estudios de Blockchains permisionadas