ECE 595: Foundations of Blockchain Systems **December 9, 2020**

Lecture 20

Lecturer: Sreeram Kannan *Scribe: Jianxiong Zhou*

Outline: In this lecture, we talk about incentives and how it plays with the rest of the system; privacy and how privacy and transparency into interact in a blockchain; some of the emerging frontiers including fairness, forensics and CAP theorem.

20.1 Incentives

20.1.1 Block reward

There are two rewards for the miners in bitcoin: transaction fee and block reward. Block reward is constant that is allocated to each block and anyone who mines a block gets the same price. This is annealed in a way called block reward schedule. Fig. 20.1 shows an example of block reward schedule. It is initially at certain price and some blocks have the same price. Then the price changes and some blocks have the same price. Block reward schedule is basically decreasing with time. To be specific, it is exponentially decreasing so that the total number of bitcoins have a finite cap. Thus as time goes on, miners have less and less rewards, which means that it is rewarding initial participants asymmetrical.

If you want to build a new cryptocurrency or any new decentralized system and you want to make sure that the system takes off, the most difficult part of building a system is enabling it to take off. A method is that you highly reward initial participation to start using the system start mining blocks at the initial stage when it's unknown whether the system will take off or not.

20.1.2 Transaction fee

But why miners keep mining when the block reward go very low? It is because the transaction fee. When each client writes down the transaction, you can also know how much they're willing to pay fees. For instance, if client A sends client B 1 million dollars, the transaction fee may be 10 dollars. And the fee goes to the miner who includes the transaction.

So miner reward goes to a certain address or they can split it to whatever addresses they want

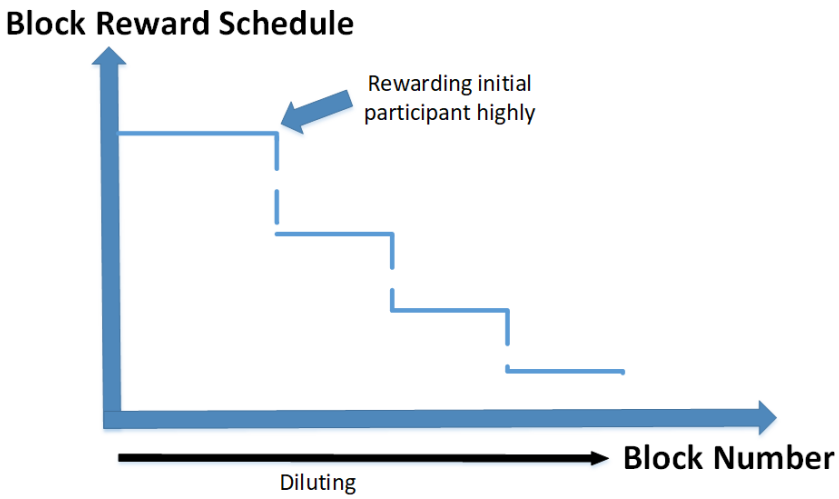

Figure 20.1: Block reward schedule.

and the total reward is basically block reward plus transaction fees. And miners are willing to choose most profitable transactions and put them in blockchain. When there is no congestion, which means there is not enough transactions to fill in block, then the transaction fee is low. But if there is high congestion, which means there are enough transactions, the transaction fee is high.

20.2 Selfish mining

Selfish mining is a method to more fraction of blocks. The core idea of selfish mining is that if there is a club public chain and you are an adversary and you mind a block. You don't release it immediately but wait till other honest blockers release one block. Then you can release two blocks together and according to the longest chain rule, you can replace others block and get more fraction of blocks. Fig. 20.2 shows the selfish mining process.

20.3 Variable difficulty

The idea of variable difficulty is to keep block mining at a fixed rate. The recalculated difficulty threshold is based on the time stamps. It is recalculated in the way that the average mining power average mining rate goes back to 10 minutes per block. With is method, selfish mining cannot gain more rewards per time.

For example, in Fig. 20.3, if in one epoch of 2014 blogs, you follow selfish mining. Since some

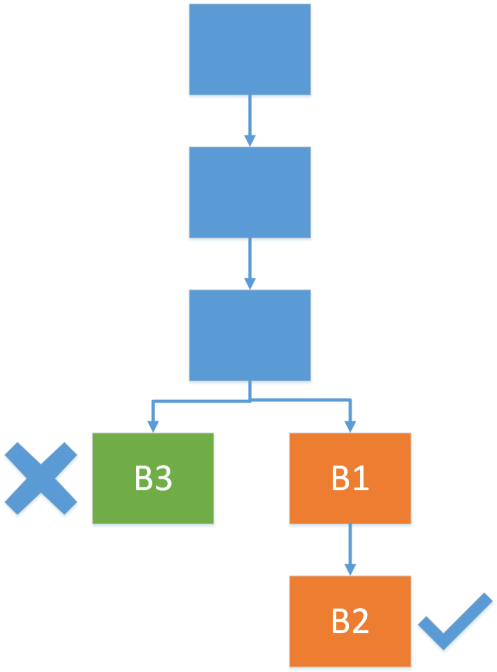

Figure 20.2: Selfish mining.

blocks get replaces in this process, it takes a longer amount of time to hit 2014 blocks, which means the difficulty is made easier for the next round. If you do not do selfish mining, you would have reached that epoch number of blocks in a shorter time. But now you are doing selfish mining and the difficulty is lower, which means everybody gets a greater fraction of rewards. Then in the next round, it is better not follow selfish mining since basically rewards became easier and you can get rewards a little bit faster than average.

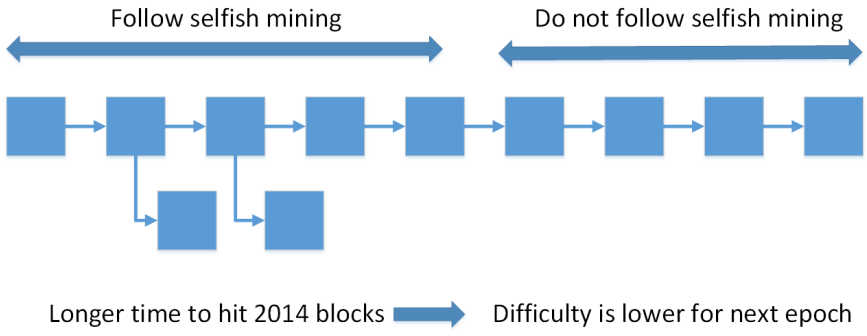

Figure 20.3: Selfish mining.

This example shows you that things which are supposedly in different layers can interact in complicated ways. Variable difficulty is for setting consensus rate about one block every 10 minutes,

but that is somehow interfering with what is happening at the higher layer and even measured in Bitcoin per unit time, the strategy can actually be more profitable.
Selfish mining can become exaggerated due to transaction fees. Suppose there are transactions which are much higher than average transaction fees and you are transferring about a billion dollars. You don't mind paying 10,000 or 100,000 transaction fees. This will create a very heavy transaction fee in certain block and it does incentive miners to construct a fork so that they can capture the transaction fee.

20.4 Fruit chains to solve incentive issues

Instead of rewarding miners who have the main blocks, you can reward miners of the transaction reward. If you are only giving block reward you can give one unit of reward to each transaction block instead of giving one unit of reward to the mining block. This is good because if the block is replaced due to the longest chain rule, transaction blocks referred by that block can also referred by the corresponding block in the main chain. In this way, even minor transactions can be included and get reward and there is no incentive to do selfish mining in this case. This method only rewards the transaction block miners, which is called sortition.
Another question about fruit chains is who gets the transaction fees. If you take a fixed number of blocks and look at all the transaction blocks miners. When it is large enough, it is basically proportional representation. You can divide the transaction fee reward of B_i among all the transaction blocks miners in the main block chain $[B_{i-L}, \ldots, B_{i-1}]$.

20.5 Transaction fees in Ethereum

Think of transactions as commands, if you execute certain number of blocks, you can get a memory state. Each command is a small program. In Ethereum you want to price programs based on how much the memory load is. The price fees are based on memory load, transaction size and CPU usage. For instance, one unit of gas can be used to have access to a certain amount of memory, a certain transaction size and a certain number of CPU cycles. If you put 100 units, then it gives you access to a certain amount of memory certain transaction size and a certain CPU usage. So a single dimension gives you a max limit on memory, transaction size and CPU usage.
If you put in a program or a contract, you can call it a command. When you are sending an

instruction, basically, you put some amount of gas. Actually it takes your instruction and then runs it through a simulator and then checks how much memory, how much transaction size and how much CPU usage is it doing. Then you know how much gas is needed to make sure this transaction can be done. You have two units to fail, one is how much gas you are putting and what is the gas price that you are willing to pay. These are deterministic since you can calculate depending on the program. If you put in more gas price that's like putting in more transaction fee. To put in less gas price that's like putting in less transactions and miners will look at the gas in the gas price before deciding which transaction. An important principle in this kind of permission as blockchain is make sure that everybody is paying for what they are costing.

20.6 Incentive attack

Incentive attack, also called "$R + \epsilon$ attack". If you want to attack the main block chain but do not have enough mining power to actually run a private attack. So what you do is that you set up a contract under the main chain and say "anybody who's building on top of this alternative chain will have a reward $R + \epsilon$", here R represents the normal reward of the blocks in main chain. If other miners are all rational players, they will go to build blocks beneath your block B_2 rather than the block B_1 in the main chain. You only need to provide them the extra ϵ reward but your blocks are quite likely to grow deeper than the main chain and replace the main chain successfully.

Even you do not have enough mining power, once the total mining power of miners who go beneath your blocks go beyond 51%, you get success. Thus incentive attack is also called "decentralized 51% attack". If the attack is successful, the cost of attack is $(k + 1)\epsilon$, here k is the depth of main chain. Fig. 20.4 gives a vivid example to show the incentive attack.

20.7 Proof of stack and economic incentives

The key idea of proof of stack is to use economic disincentives, which means if somebody misbehaves be detected, their stake will be token away as capital punishment. If there is an attack, usually a safety violation, we need algorithms to allow others to pinpoint some fraction of the stake to point out some fraction of nodes as adversarial. Unless the attacker is willing to burn a large fraction of the stake, they will not be able to execute this attack. There is a bunch of papers

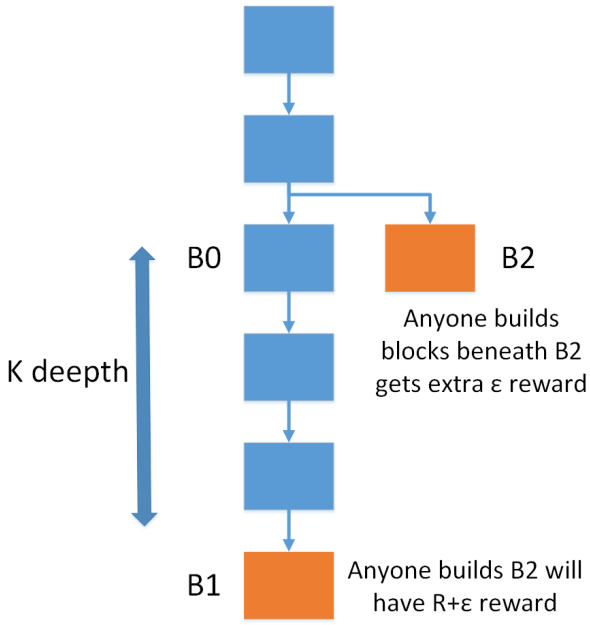

Figure 20.4: Incentive attack.

like Casper and CBC that came up with this series of protocols which have this kind of property. In general, what we do is that we created a family of protocols. We call this BFD forensics. Forensics means that if something bad happened, you are finding out who did it. Many classic algorithms also allow such detection.

20.8 Front-running

For instance, if Warren Buffett has made an order on selling Southwest Airlines stock and you know this may crash Southwest Airlines. So you want to front run your Southwest Airlines stock before Warren Buffett's order hits the market. Suppose Warren Buffett's transaction is called t_1 and your transaction is called t_2. In New York Stock Exchange, t_1 is always head of t_2. But in block chain, it takes time for t_1 to get included in a block and even more time. So you can increase your transaction fees, making it much higher than that of t_1, to attract more miners to includes your transaction first. This will change the order, which is called "non-fair ordering".
"Non-fair ordering" can be a big problem of block chain. Thus Fair Ordered Protocol is created to ensure that the transaction order reflects the order of arrival.