On unstabilities of the Bitcoin protocol

Ricardo Pérez-Marco (CNRS, IMJ-PRG, Paris 7)

DLT Workshop

Perugia

February 1, 2018

(*Bitcoin and Decentralized Trust Protocols*, Newsletter of the European Math. Soc., 100, June 2016. ArXiv 1601.05254)

Contents

Bitcoin paper

S. Nakamoto, November 1st 2008,

Bitcoin paper

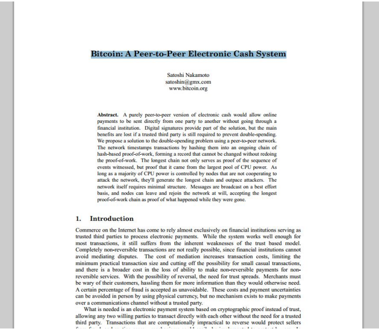
S. Nakamoto, November 1st 2008,

“Bitcoin: A peer-to-peer electronic cash system”

Bitcoin paper

S. Nakamoto, November 1st 2008,

“Bitcoin: A peer-to-peer electronic cash system”



Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for non-reversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers

Transparency Theorem

- Bitcoin is an electronic currency modeled by gold.

Transparency Theorem

- Bitcoin is an electronic currency modeled by gold.
- Bitcoin does not depend on any central authority.

Transparency Theorem

- Bitcoin is an electronic currency modeled by gold.
- Bitcoin does not depend on any central authority.
- Bitcoin protocol runs on open software.

Transparency Theorem

- Bitcoin is an electronic currency modeled by gold.
- Bitcoin does not depend on any central authority.
- Bitcoin protocol runs on open software.
- To avoid the “**double spend problem**” Bitcoin relies on a public ledger.

Transparency Theorem

- Bitcoin is an electronic currency modeled by gold.
- Bitcoin does not depend on any central authority.
- Bitcoin protocol runs on open software.
- To avoid the “**double spend problem**” Bitcoin relies on a public ledger. This is general and necessary:

Transparency Theorem

- Bitcoin is an electronic currency modeled by gold.
- Bitcoin does not depend on any central authority.
- Bitcoin protocol runs on open software.
- To avoid the "**double spend problem**" Bitcoin relies on a public ledger. This is general and necessary:

Theorem

Transparency Theorem: *An electronic decentralized currency must rely on a public ledger.*

The blockchain

- The public ledger is an incorruptible public database of all transactions called “**the blockchain**”.

The blockchain

- The public ledger is an incorruptible public database of all transactions called “**the blockchain**”.
- Anyone can write in the blockchain.

The blockchain

- The public ledger is an incorruptible public database of all transactions called "**the blockchain**".
- Anyone can write in the blockchain.
- Anyone can have a copy of the blockchain.

The blockchain

- The public ledger is an incorruptible public database of all transactions called "**the blockchain**".
- Anyone can write in the blockchain.
- Anyone can have a copy of the blockchain.
- The blockchain is composed by a chronological sequence of cryptological chained blocks.

The blockchain

• The public ledger is an incorruptible public database of all transactions called “**the blockchain**”.

• Anyone can write in the blockchain.

• Anyone can have a copy of the blockchain.

• The blockchain is composed by a chronological sequence of cryptological chained blocks.

• Each block contains a set of transactions.

The blockchain

- The public ledger is an incorruptible public database of all transactions called “**the blockchain**”.
- Anyone can write in the blockchain.
- Anyone can have a copy of the blockchain.
- The blockchain is composed by a chronological sequence of cryptological chained blocks.
- Each block contains a set of transactions.
- Each new block is generated in about 10 minutes.

The blockchain

• The public ledger is an incorruptible public database of all transactions called “**the blockchain**”.

• Anyone can write in the blockchain.

• Anyone can have a copy of the blockchain.

• The blockchain is composed by a chronological sequence of cryptological chained blocks.

• Each block contains a set of transactions.

• Each new block is generated in about 10 minutes.

• The blocks are generated by “miners” that validate current transactions.

The Trust Machine

- The core of the Bitcoin protocol is the algorithm to ensure that this database cannot be forged.

The Trust Machine

- The core of the Bitcoin protocol is the algorithm to ensure that this database cannot be forged.
- The mechanism of consensus: “The trust machine”.

Nodes

- The Bitcoin Network is composed by nodes that communicate with each other.

Nodes

- The Bitcoin Network is composed by nodes that communicate with each other.
- Nodes check and broadcast transactions.

Nodes

- The Bitcoin Network is composed by nodes that communicate with each other.
- Nodes check and broadcast transactions.
- Some nodes are miners that validate transactions.

Nodes

- The Bitcoin Network is composed by nodes that communicate with each other.
- Nodes check and broadcast transactions.
- Some nodes are miners that validate transactions.
- Anyone can join and participate in the network.

Nodes

- The Bitcoin Network is composed by nodes that communicate with each other.
- Nodes check and broadcast transactions.
- Some nodes are miners that validate transactions.
- Anyone can join and participate in the network.
- To avoid Sybil attacks a “Proof of Work” (PoW) for miners is required.

Nodes

- The Bitcoin Network is composed by nodes that communicate with each other.
- Nodes check and broadcast transactions.
- Some nodes are miners that validate transactions.
- Anyone can join and participate in the network.
- To avoid Sybil attacks a “Proof of Work” (PoW) for miners is required.

Reaching consensus

- How to reach consensus in a network with insecure communications and malicious nodes but a majority of honest agents?

Reaching consensus

- How to reach consensus in a network with insecure communications and malicious nodes but a majority of honest agents?

The Byzantine Generals Problem.
The situation can be described as the siege of a city by a group of generals of the Byzantine army. Communicating only by messenger, the generals must agree upon a common battle plan. However, one or more of them may be traitors who will try to confuse the others. The problem is to find an algorithm to ensure that the loyal generals will reach an agreement.

Reaching consensus

- How to reach consensus in a network with insecure communications and malicious nodes but a majority of honest agents?

The Byzantine Generals Problem.
The situation can be described as the siege of a city by a group of generals of the Byzantine army. Communicating only by messenger, the generals must agree upon a common battle plan. However, one or more of them may be traitors who will try to confuse the others. The problem is to find an algorithm to ensure that the loyal generals will reach an agreement.

- **Nakamoto Byzantine Generals Problem:** The number of generals is not fixed.

Reaching consensus

- How to reach consensus in a network with insecure communications and malicious nodes but a majority of honest agents?

The Byzantine Generals Problem.
The situation can be described as the siege of a city by a group of generals of the Byzantine army. Communicating only by messenger, the generals must agree upon a common battle plan. However, one or more of them may be traitors who will try to confuse the others. The problem is to find an algorithm to ensure that the loyal generals will reach an agreement.

- **Nakamoto Byzantine Generals Problem:** The number of generals is not fixed.

Reaching Consensus

- The idea is to select randomly who validates the next block of transactions.

Reaching Consensus

- The idea is to select randomly who validates the next block of transactions.
- A decentralized “lottery” is set by the PoW.

Reaching Consensus

- The idea is to select randomly who validates the next block of transactions.
- A decentralized “lottery” is set by the PoW.
- A computationally intensive problem is set to validate a block.

Reaching Consensus

- The idea is to select randomly who validates the next block of transactions.
- A decentralized “lottery” is set by the PoW.
- A computationally intensive problem is set to validate a block.
- The problem is difficult to solve, but the solution is easy to check.

Reaching Consensus

- The idea is to select randomly who validates the next block of transactions.
- A decentralized “lottery” is set by the PoW.
- A computationally intensive problem is set to validate a block.
- The problem is difficult to solve, but the solution is easy to check.
- The difficulty is adjusted to find a solution in about 10 minutes.

Reaching Consensus

- The idea is to select randomly who validates the next block of transactions.
- A decentralized “lottery” is set by the PoW.
- A computationally intensive problem is set to validate a block.
- The problem is difficult to solve, but the solution is easy to check.
- The difficulty is adjusted to find a solution in about 10 minutes.
- The miner that solves it receives an award in newly created bitcoins.

Decentralized governance

- There cannot be a centralized authority that enforces the protocol

Decentralized governance

- There cannot be a centralized authority that enforces the protocol
- Nobody can coerce actors to follow the protocol.

Decentralized governance

- There cannot be a centralized authority that enforces the protocol
- Nobody can coerce actors to follow the protocol.
- The only possible decentralized governance is to align the protocol rules with self-interest of the participants.

Decentralized governance

• There cannot be a centralized authority that enforces the protocol

• Nobody can coerce actors to follow the protocol.

• The only possible decentralized governance is to align the protocol rules with self-interest of the participants.

This is extremely hard!

Decentralized governance

• There cannot be a centralized authority that enforces the protocol

• Nobody can coerce actors to follow the protocol.

• The only possible decentralized governance is to align the protocol rules with self-interest of the participants.

This is extremely hard!

It is a miracle that the Bitcoin protocol works!

Decentralized governance

• There cannot be a centralized authority that enforces the protocol

• Nobody can coerce actors to follow the protocol.

• The only possible decentralized governance is to align the protocol rules with self-interest of the participants.

This is extremely hard!

It is a miracle that the Bitcoin protocol works!

51% attack

51% attack

- The protocol cannot function if anyone controls more than 50% of the hashrate.

51% attack

- The protocol cannot function if anyone controls more than 50% of the hashrate.

Double spend attacks, control of the blocks that are included in the blockchain, etc

51% attack

- The protocol cannot function if anyone controls more than 50% of the hashrate.

Double spend attacks, control of the blocks that are included in the blockchain, etc

- Decentralized mining is fundamental to avoid a 51% attack

51% attack

• The protocol cannot function if anyone controls more than 50% of the hashrate.

Double spend attacks, control of the blocks that are included in the blockchain, etc

• Decentralized mining is fundamental to avoid a 51% attack

• Big pools are a thread to mining decentralization.

51% attack

- The protocol cannot function if anyone controls more than 50% of the hashrate.

Double spend attacks, control of the blocks that are included in the blockchain, etc

- Decentralized mining is fundamental to avoid a 51% attack
- Big pools are a thread to mining decentralization.
- Monopole position on mining hardware manufacturing is a thread to mining decentralization.

Selfish mining attack

(join work with C. Grunspan)

• A protocole rule is that miners release the mined blocks as soon as they are mined.

Selfish mining attack

(join work with C. Grunspan)

• A protocole rule is that miners release the mined blocks as soon as they are mined.

• Self-interested seems to be well aligned with this rule because of the block reward.

Selfish mining attack

(join work with C. Grunspan)

• A protocole rule is that miners release the mined blocks as soon as they are mined.

• Self-interested seems to be well aligned with this rule because of the block reward.

• A block witholding strategy allows to invalidate blocks of miner competitors.

Selfish mining attack

(join work with C. Grunspan)

• A protocole rule is that miners release the mined blocks as soon as they are mined.

• Self-interested seems to be well aligned with this rule because of the block reward.

• A block witholding strategy allows to invalidate blocks of miner competitors.

• This strategy is possible with less than 50% hashrate.

Selfish mining attack

(join work with C. Grunspan)

• A protocole rule is that miners release the mined blocks as soon as they are mined.

• Self-interested seems to be well aligned with this rule because of the block reward.

• A block witholding strategy allows to invalidate blocks of miner competitors.

• This strategy is possible with less than 50% hashrate.

• Costs of this strategy are not properly accounted in the literature.

Profitability of selfish mining

- Selfish mining strategy is not profitable without an adjustment of the difficulty.

Profitability of selfish mining

- Selfish mining strategy is not profitable without an adjustment of the difficulty.
- The self-mining attack slows down the network and block validation.

Profitability of selfish mining

- Selfish mining strategy is not profitable without an adjustment of the difficulty.
- The self-mining attack slows down the network and block validation.
- More precisely, $P\&L$ of the selfish mining strategy is negative if the difficulty does not adjust.

Profitability of selfish mining

- Selfish mining strategy is not profitable without an adjustment of the difficulty.
- The self-mining attack slows down the network and block validation.
- More precisely, $P\&L$ of the selfish mining strategy is negative if the difficulty does not adjust.
- The profitability of the selfish-mining strategy relies crucially on the good connection to the network.

Profitability of selfish mining

- Selfish mining strategy is not profitable without an adjustment of the difficulty.
- The self-mining attack slows down the network and block validation.
- More precisely, $P\&L$ of the selfish mining strategy is negative if the difficulty does not adjust.
- The profitability of the selfish-mining strategy relies crucially on the good connection to the network.
- Only viable with more than $30 - 40\%$ of the hashrate.

Selfish-mining and Nash equilibrium

- Why we don't see selfish mining in the network?

Selfish-mining and Nash equilibrium

- Why we don't see selfish mining in the network?
- Selfish mining is only profitable if there is only one bad actor.

Selfish-mining and Nash equilibrium

- Why we don't see selfish mining in the network?
- Selfish mining is only profitable if there is only one bad actor.
- Nash equilibrium: It is in the interest of all the miners to not start a selfish mining war because the network will stall.

Selfish-mining and Nash equilibrium

- Why we don't see selfish mining in the network?
- Selfish mining is only profitable if there is only one bad actor.
- Nash equilibrium: It is in the interest of all the miners to not start a selfish mining war because the network will stall.
- After all, the protocol is well aligned.

Catch-up mining

(join work with C. Grunspan)

• Protocol rule: Miners should mine on top of the public blockchain.

Catch-up mining

(join work with C. Grunspan)

• Protocol rule: Miners should mine on top of the public blockchain.

• So, if they mine a block at the same time that a new block is found they should abandon their block and start mining on top of the new block.

Catch-up mining

(join work with C. Grunspan)

• Protocol rule: Miners should mine on top of the public blockchain.

• So, if they mine a block at the same time that a new block is found they should abandon their block and start mining on top of the new block.

• The best self-interest strategy is to release the block and start mining on top of it to try to orphan the last public block (the protocol is slightly misaligned with self-interest at this point).

Catch-up mining

(join work with C. Grunspan)

• Protocol rule: Miners should mine on top of the public blockchain.

• So, if they mine a block at the same time that a new block is found they should abandon their block and start mining on top of the new block.

• The best self-interest strategy is to release the block and start mining on top of it to try to orphan the last public block (the protocol is slightly misaligned with self-interest at this point).

• But if a second block is released in the public network, with 1-block behind it seems intuitively clear that the miner should adopt the public blockchain and discard his block.

Catch-up mining

(join work with C. Grunspan)

• Protocol rule: Miners should mine on top of the public blockchain.

• So, if they mine a block at the same time that a new block is found they should abandon their block and start mining on top of the new block.

• The best self-interest strategy is to release the block and start mining on top of it to try to orphan the last public block (the protocol is slightly misaligned with self-interest at this point).

• But if a second block is released in the public network, with 1-block behind it seems intuitively clear that the miner should adopt the public blockchain and discard his block.

- It may make sense to catch-up mining because although the probability of success is small, the reward is high since it reaps all the rewards of the invalidated blocks

- It may make sense to catch-up mining because although the probability of success is small, the reward is high since it reaps all the rewards of the invalidated blocks
- Catch-up mining is only profitable for a high hashrate.

- It may make sense to catch-up mining because although the probability of success is small, the reward is high since it reaps all the rewards of the invalidated blocks
- Catch-up mining is only profitable for a high hashrate.
- It is a complex mathematical problem.

- It may make sense to catch-up mining because although the probability of success is small, the reward is high since it reaps all the rewards of the invalidated blocks
- Catch-up mining is only profitable for a high hashrate.
- It is a complex mathematical problem.
- Equivalent to a gambling problem: We have a lag of m and we play n rounds of biased coin flipping heads (probability $q < 1/2$) or tails (probability $p = 1 - q$). Reward v if we catch-up before n rounds. Each time we have a losing round the reward increases by 1.

- It may make sense to catch-up mining because although the probability of success is small, the reward is high since it reaps all the rewards of the invalidated blocks
- Catch-up mining is only profitable for a high hashrate.
- It is a complex mathematical problem.
- Equivalent to a gambling problem: We have a lag of m and we play n rounds of biased coin flipping heads (probability $q < 1/2$) or tails (probability $p = 1 - q$). Reward v if we catch-up before n rounds. Each time we have a losing round the reward increases by 1.
- $E_m^n(v)$ expected value for the optimal strategy for catching-up a lag of m in n rounds with a reward v.

- It may make sense to catch-up mining because although the probability of success is small, the reward is high since it reaps all the rewards of the invalidated blocks
- Catch-up mining is only profitable for a high hashrate.
- It is a complex mathematical problem.
- Equivalent to a gambling problem: We have a lag of m and we play n rounds of biased coin flipping heads (probability $q < 1/2$) or tails (probability $p = 1 - q$). Reward v if we catch-up before n rounds. Each time we have a losing round the reward increases by 1.
- $E_m^n(v)$ expected value for the optimal strategy for catching-up a lag of m in n rounds with a reward v.
- $v \mapsto E_m^n(v)$ is non-decreasing.

- It may make sense to catch-up mining because although the probability of success is small, the reward is high since it reaps all the rewards of the invalidated blocks
- Catch-up mining is only profitable for a high hashrate.
- It is a complex mathematical problem.
- Equivalent to a gambling problem: We have a lag of m and we play n rounds of biased coin flipping heads (probability $q < 1/2$) or tails (probability $p = 1 - q$). Reward v if we catch-up before n rounds. Each time we have a losing round the reward increases by 1.
- $E_m^n(v)$ expected value for the optimal strategy for catching-up a lag of m in n rounds with a reward v.
- $v \mapsto E_m^n(v)$ is non-decreasing.
- There is a unique $v_m^n = (E_m^n)^{-1}(0)$.

Generalized Dyck paths

Let (m, n) be given. A function $f : [0, n] \to \mathbb{N}$ is a (m, n)-Dyck path if

Generalized Dyck paths

Let (m, n) be given. A function $f : [0, n] \to \mathbb{N}$ is a (m, n)-Dyck path if

1. $f(0) = m$

Generalized Dyck paths

Let (m, n) be given. A function $f : [0, n] \to \mathbb{N}$ is a (m, n)-Dyck path if

1. $f(0) = m$

Generalized Dyck paths

Let (m, n) be given. A function $f : [0, n] \to \mathbb{N}$ is a (m, n)-Dyck path if

1. $f(0) = m$
2. $f(n) = 0$

Generalized Dyck paths

Let (m, n) be given. A function $f : [0, n] \to \mathbb{N}$ is a (m, n)-Dyck path if

1. $f(0) = m$
2. $f(n) = 0$

Generalized Dyck paths

Let (m, n) be given. A function $f : [0, n] \to \mathbb{N}$ is a (m, n)-Dyck path if

1. $f(0) = m$
2. $f(n) = 0$
3. For $k < n$, $f(k) > 0$

Generalized Dyck paths

Let (m, n) be given. A function $f : [0, n] \to \mathbb{N}$ is a (m, n)-Dyck path if

1. $f(0) = m$
2. $f(n) = 0$
3. For $k < n$, $f(k) > 0$

Generalized Dyck paths

Let (m, n) be given. A function $f : [0, n] \to \mathbb{N}$ is a (m, n)-Dyck path if

1. $f(0) = m$
2. $f(n) = 0$
3. For $k < n$, $f(k) > 0$
4. For $k < n$, $|f(k+1) - f(k)| = 1$

Generalized Dyck paths

Let (m, n) be given. A function $f : [0, n] \to \mathbb{N}$ is a (m, n)-Dyck path if

1. $f(0) = m$
2. $f(n) = 0$
3. For $k < n$, $f(k) > 0$
4. For $k < n$, $|f(k+1) - f(k)| = 1$

Generalized Dyck paths

Let (m, n) be given. A function $f : [0, n] \to \mathbb{N}$ is a (m, n)-Dyck path if

1. $f(0) = m$
2. $f(n) = 0$
3. For $k < n$, $f(k) > 0$
4. For $k < n$, $|f(k+1) - f(k)| = 1$

Generalized Dyck paths

Let (m, n) be given. A function $f : [0, n] \to \mathbb{N}$ is a (m, n)-Dyck path if

1. $f(0) = m$
2. $f(n) = 0$
3. For $k < n$, $f(k) > 0$
4. For $k < n$, $|f(k+1) - f(k)| = 1$

• We denote $|f| = n$ (the length of f), and $\mathcal{D}_m^n$ the set of all $(m.k)$-Dyck paths f with $|f| = k < n$.

Generalized Dyck paths

Let (m, n) be given. A function $f : [0, n] \to \mathbb{N}$ is a (m, n)-Dyck path if

1. $f(0) = m$
2. $f(n) = 0$
3. For $k < n$, $f(k) > 0$
4. For $k < n$, $|f(k+1) - f(k)| = 1$

• We denote $|f| = n$ (the length of f), and $\mathcal{D}_m^n$ the set of all $(m.k)$-Dyck paths f with $|f| = k < n$.
• $f(k)$ is the lag after k turns.

Generalized Dyck paths

Let (m, n) be given. A function $f : [0, n] \to \mathbb{N}$ is a (m, n)-Dyck path if

1. $f(0) = m$
2. $f(n) = 0$
3. For $k < n$, $f(k) > 0$
4. For $k < n$, $|f(k+1) - f(k)| = 1$

• We denote $|f| = n$ (the length of f), and $\mathcal{D}_m^n$ the set of all $(m.k)$-Dyck paths f with $|f| = k < n$.
• $f(k)$ is the lag after k turns.
• $\mu_f(k) = \sum_{j=0}^{k-1} (f(k+1) - f(k))_+$ favorable rounds in k turns.

Generalized Dyck paths

Let (m, n) be given. A function $f : [0, n] \to \mathbb{N}$ is a (m, n)-Dyck path if

1. $f(0) = m$
2. $f(n) = 0$
3. For $k < n$, $f(k) > 0$
4. For $k < n$, $|f(k+1) - f(k)| = 1$

• We denote $|f| = n$ (the length of f), and $\mathcal{D}_m^n$ the set of all $(m.k)$-Dyck paths f with $|f| = k < n$.
• $f(k)$ is the lag after k turns.
• $\mu_f(k) = \sum_{j=0}^{k-1} (f(k+1) - f(k))_+$ favorable rounds in k turns.
• $w(f) = \sup_{k<|f|} v_{f(k)}^n - \mu_f(k)$ minimal reward allowing to continue playing.

Generalized Dyck paths

Let (m, n) be given. A function $f : [0, n] \to \mathbb{N}$ is a (m, n)-Dyck path if

1. $f(0) = m$
2. $f(n) = 0$
3. For $k < n$, $f(k) > 0$
4. For $k < n$, $|f(k+1) - f(k)| = 1$

• We denote $|f| = n$ (the length of f), and $\mathcal{D}_m^n$ the set of all $(m.k)$-Dyck paths f with $|f| = k < n$.
• $f(k)$ is the lag after k turns.
• $\mu_f(k) = \sum_{j=0}^{k-1} (f(k+1) - f(k))_+$ favorable rounds in k turns.
• $w(f) = \sup_{k<|f|} v_{f(k)}^n - \mu_f(k)$ minimal reward allowing to continue playing.
• $\pi(f) = p^{\mu_f(|f|)} q^{|f|-\mu_f(|f|)}$ probability of the path f.

Generalized Dyck paths

Let (m, n) be given. A function $f : [0, n] \to \mathbb{N}$ is a (m, n)-Dyck path if

1. $f(0) = m$
2. $f(n) = 0$
3. For $k < n$, $f(k) > 0$
4. For $k < n$, $|f(k+1) - f(k)| = 1$

• We denote $|f| = n$ (the length of f), and $\mathcal{D}_m^n$ the set of all $(m.k)$-Dyck paths f with $|f| = k < n$.
• $f(k)$ is the lag after k turns.
• $\mu_f(k) = \sum_{j=0}^{k-1} (f(k+1) - f(k))_+$ favorable rounds in k turns.
• $w(f) = \sup_{k<|f|} v_{f(k)}^n - \mu_f(k)$ minimal reward allowing to continue playing.
• $\pi(f) = p^{\mu_f(|f|)} q^{|f|-\mu_f(|f|)}$ probability of the path f.

Summation formula

Theorem (Formula with generalized Dyck paths)

$$E_m^n(v) = \sum_{f \in \mathcal{D}_m^n} \pi(f)(f - w(f))_+$$

Summation formula

Theorem (Formula with generalized Dyck paths)

$$E_m^n(v) = \sum_{f \in \mathcal{D}_m^n} \pi(f)(f - w(f))_+$$

Summation formula

Theorem (Formula with generalized Dyck paths)

$$E_m^n(v) = \sum_{f \in \mathcal{D}_m^n} \pi(f)(f - w(f))_+$$

Practical application

Practical application

Theorem

If $q > 0.43$, $m = 2$, and $b > 0$ is the block reward, then $\lim_{n \to +\infty} E_n^2(3b) > 0$.

Practical application

Theorem

If $q > 0.43$, $m = 2$, *and* $b > 0$ *is the block reward, then* $\lim_{n \to +\infty} E_n^2(3b) > 0$.

Practical application

Theorem

If $q > 0.43$, $m = 2$, and $b > 0$ is the block reward, then $\lim_{n \to +\infty} E_n^2(3b) > 0$.

This means that it makes sense to catch-up mining 2 blocks behind if your hashrate is over 43%.

Practical application

Theorem

If $q > 0.43$, $m = 2$, and $b > 0$ is the block reward, then $\lim_{n \to +\infty} E_n^2(3b) > 0$.

This means that it makes sense to catch-up mining 2 blocks behind if your hashrate is over 43%.

The bitcoin protocol is unstable with respect to catch-up mining.

Sorry for the formulas...

Sorry for the formulas...

Sorry for the formulas...

...and thank you for your attention!!