Ansible, practical faster-than-light secure 0-conf transactions for Bitcoin

By Joannes Vermorel, April 5nd, 2018

The lightning lost the race to the ansible; speed of light wasn't nearly fast enough.

Document status: EARLY DRAFT

In this document, Bitcoin always refers to Bitcoin Cash.

Abstract: The Ansible is a peer-to-peer pre-consensus signal that, by itself, makes 0-conf transactions secure. The Ansible is a self-fulfilling prophecy because making it so aligns with the economic interests of the miners who have to remain competitive. The Ansible confers two seemingly counterintuitive properties to the Nakamoto consensus. First, securing 0-conf transactions do not require any kind of retaliation against Byzantine miners, thus no change to the Nakamoto consensus. Second, 0-conf transactions can be secured with arbitrarily low latencies on earth, despite the fact that this proposition appears to violate the speed of light. The Ansible perspective clarifies why larger and infrequent blocks are actually highly desirable to secure 0-conf transactions.

Overview

Satoshi Nakamoto (2008) has introduced Bitcoin as a protocol to form consensus over a distributed peer-to-peer network, the *Nakamoto* consensus delivered through the blockchain. The blockchain technology provides a decentralized, open, Byzantine fault-tolerant transaction mechanism. In its original form, the Nakamoto consensus delivers secure transactions (irreversible) but a high latency, typically 10 min or more on average.

The original Bitcoin paper does not clarify how the same properties can be reproduced with a much lower latency, to make Bitcoin compatible with more stringent real-world use cases where any latency beyond about 3 seconds would be deemed impractical by users.

From the Bitcoin perspective, this problem is known as the 0-conf problem, i.e. securing transactions that have not yet been included in a valid block as defined by the Nakamoto consensus. Securing 0-conf transactions has been extensively studied, and arguably many of the most notable propositions involve modifying the Nakamoto consensus, as illustrated by the Bitcoin NG proposal.

The present paper takes a conservative stance on the Nakamoto consensus. This consensus is indeed the only scheme of its class (decentralized, fault-tolerant, etc) that has reliably demonstrated its resilience, for almost a decade, to relentless attacks.

Thus, any replacement for the Nakamoto consensus comes with a rather drastic burden of proof, as it cannot realistically reproduce a decade of real-world operational success at both defeating attacks while not generating entire classes of unintended problems in its trail.

Thus, the Ansible, the approach proposed in the present article, does not attempt to modify the Nakamoto consensus. Instead, the Ansible presents itself as a pre-consensus signal.

Let's define a pre-consensus *signal* (simply "signal" in the following) as a set of data distributed to the Bitcoin network that represents a non-ambiguous statement about the transactions that are *expected* to be included in the next block. For any possible transaction, the signal carries a binary statement: either the transaction will make its way into the next block, or it won't.

The signal is also characterized by its *operating latency* which is the minimal latency that can be reliably relied upon by external observers, defined as unprivileged participants of the signal propagation itself. E.g. if a given signal has an operating latency of 500 ms, then it means a Bitcoin payee can trust the 0-conf transactions that are being given to him within 500 ms.

Unlike a pre-consensus *mechanism* which is implicitly associated with an intent of rejecting what would otherwise be a valid transaction or a valid block according to the Nakamoto consensus, a pre-consensus signal is, as the name suggests, a pure matter of making some data of the Bitcoin network *provably* and *diligently* observable. A pre-consensus signal claims nothing beyond its own existence being made accessible to all Bitcoin participants who are willing to join the signal itself, miners and users alike.

A pre-consensus signal can be attacked by two classes of adversaries:

- Byzantine miners - who have control on the production of blocks, but only limited control on the timing of the production of said blocks.
- Byzantine users - who have control of the timing of the production of their transactions, but who do not have the capacity to produce blocks.

A *bad transaction* refers to a transaction published before consensus with the intent of conflicting the signal, as an attempt to fool the payee into placing a misguided trust into the accuracy (see below) of the signal.

A *poison transaction* refers to a transaction published before the consensus which did get included within a block while defeating the signal for a duration longer than the expected operating latency of the signal.

As the signal acts as a binary classification function applied to 0-conf Bitcoin transactions, it is characterized by two measurements that define its performance: its *sensibility* and its *specificity*. The sensibility is the positive rate of detection of transactions as correct candidates for inclusion in the next block. The specificity is the negative rate of detection of transactions that are correct non-candidates for inclusion in the next block. From a practical perspective of securing the 0-conf, only the sensibility matters, as a highly sensible signal has to be a highly specific one as well, precisely due to the mutually-exclusive nature of bad transactions. Thus, in the following, we will use the term *accuracy* (while referring instead to the *sensibility*) to refer to the overall performance of a given signal, implicitly acknowledging that sensibility and specificity measurements collapse into a single value.

The purpose of a pre-consensus signal is to be as accurate and as diligent as possible in its statements. If a signal can achieve a 100% accuracy in an operating latency of 1 second, then 0-conf transactions are secure after 1 second for any Bitcoin payee because they use the signal itself to assess the validity of the claim made by their Bitcoin payers.

However, even if a signal does not achieve a perfect accuracy, but merely a very high accuracy, say 99.9% or more, the signal has obvious value for Bitcoin payees in aggregate. Indeed, by the very definition of the signal accuracy, a 99.9% accurate signal indicates to payees, in aggregate, that their trust in the signal only exposes them to a 0.1% chance of being defrauded. Thus, payees can, in aggregate, trust 0-conf transactions, granted they are willing to absorb a 0.1% fraud risk, which can be done by internalizing the cost associated to this risk into their prices, as communicated to payers.

One key insight of pre-consensus signaling is that secure 0-conf transactions merely require the existence of reasonably accurate signals. If such a signal can be exhibited, then, 0-conf transactions are *practically* secure from a Bitcoin payee perspective.

RBF bribes are more risky than they appear

A dishonest payer who intends to execute a double-spend fraud requires the support of a dishonest miner to execute its fraud. The payer can bribe the miner by communicating directly to the miner - or even by managing to achieve a public broadcast of the bad transaction - to give the miner an incentive to cooperate in the fraud. In the Bitcoin community, this specific fraud is known under the name of RBF (replace-by-fee).

As a tangential note, the author would like to point out that the notion of *fraud* is dependent on the underlying social contract. Indeed, Bitcoin is best described as a *social contract* despite the fact that, unlike pretty much any other social contract found in human history, Bitcoin needs computing hardware to be executed at all. If the social contract of Bitcoin is generally

understood with a caveat stating that *poison transactions* are fraud, then they are[1], precisely because they betray the underlying social contract. Then, this very social contract is amenable to change as well. If a competitive fork of Bitcoin establish RBF as a first-class citizen and declares RBF to be an intended feature, then the situation is different. In this alternative non-Bitcoin social contract, participants are deemed honest when they exploit the RBF feature; thus, in this non-Bitcoin context RBF is *not* fraud.

Back to Bitcoin, a misguided interpretation of the social contract could lead to an understanding where a rational but dishonest miner would willingly engage in RBF frauds, as the miner seems to make an easy profit from them.

Let's immediately point out that taking the fee and successfully defrauding the payee would make the miner an accomplice of the fraud in every court of justice[2]. This point alone is a cause of concern for the miner, because suddenly, it is no more merely a dishonorable company frowned upon, but plain *organized crime*, which is actively fought by authorities. Then, the other miners themselves, the honest ones playing the long game, would probably sever their network ties, blacklisting the dishonest miner.

Then, there is an even greater cause of concern for a miner who would be fine being organized crime: *the miner doesn't know who is being defrauded*. Unlike the dishonest payer who can - maybe - remain anonymous, this option is not readily available to the miner. Miners have to build or buy, and then operate all their mining hardware. In the real world, everything leaves tracks and witnesses. It's much easier to hide a metric ton of cocaine than to hide 1MW of sustained energy consumption.

Also, miners compete on block propagation latency. Hiding behind Tor to maintain anonymity incurs a massive latency penalty. Only a steady stream of bribes and frauds would compensate for the economic inefficiencies associated with a fully anonymous miner.

By taking bribes at random, the miner would also make random enemies. Seeking retaliation against overtly hostile behavior is a very human trait. A small fraction of the dishonest users would most likely attempt to defraud all the wrong people, e.g. the IRS in the USA for example, if the IRS were to ever to accept Bitcoin as payment. Thus, a rational yet dishonest miner would have to *vet* somehow the frauders themselves, to keep this runaway generation of enemies under control. The author claims no expertise in running an *organized crime* organization, but

---

[1] The curious reader might consider why not paying the bill after eating at restaurant table is theft. Indeed, there is no written contract signed. There is not even a verbal consent given by the patron. The social contract is only given by the context and the social norm. If you sit at a restaurant table and order anything, the society at large expects you to pay for what you just ordered. Failure to comply is deemed theft.

[2] Fraudsters are inventive. Courts of justice have a long track record of being good at as qualifying as fraud ingenious fraudsters who invariably attempt to disguise their fraud as an acceptable edge-case of the social contract. In case of lingering doubts, the author suggests the reader to watch the movie Chicago https://en.wikipedia.org/wiki/Chicago_(2002_film)

casual observations indicate that lack of trust appears to be an endemic problem within those organizations. Running a viable frauder-vetting scheme at scale - while remaining fully anonymous - appears improbable.

If it ain't broke, don't fix it

Unless the overall economic damage of poison transactions can be demonstrated to be sizeable, there is no point in attempting to modify the Nakamoto consensus while trying to improve the security of 0-conf transactions. Indeed, any alternative consensus carries the risk of generating unintended effects, which have to be weighted against positively observable flaws of the Nakamoto consensus.

Indeed, as *you cannot optimize what you cannot measure*, miners who have stakes in preserving an otherwise functional system should reject any attempt to modify the Nakamoto consensus to secure 0-conf transaction unless there is a convincing rationale backing the change being proposed.

As securing 0-conf transaction requires a signal to establish whether there is a problem in the first place, a signal must be built as a preliminary low-risk step, if only to assess the situation. Thus, if the Bitcoin community intends to secure 0-conf transactions, the next rational step is to have the Bitcoin community build a signal, if only as an intellectual demonstration of the validity of the claim that something remains to be done to make those 0-conf transaction secure.

Self-fulfilling prophetic effectiveness of an accurate signal

Rational miners who intend to be competitive require a signal as accurate as possible. Indeed, having access to an accurate signal is of high practical importance because it lets them process faster (more on that below) any incoming block found by another miner, precisely because of the very accuracy of the signal itself. Then, as a miner knows that its competitors, being good at *competing*, will rationally seek this very competitive edge as well, the miner can trust that emitting a block fully aligned with the signal itself will *propagate faster,* hence securing another competitive edge.

A miner who has access to a more accurate signal can validate blocks faster because:

- all the CPU intensive cryptographic operations can be done ahead of time
- all the I/O intensive UTXO operations can be done ahead of time (to be reverted only for poison transactions).

Based on the *faster block processing claim* (just demonstrated), we have almost completed the demonstration of the self-prophetic effectiveness of an accurate signal. The one remaining gap in this demonstration is that a rational but dishonest miner remains capable of closely following the signal, except for his very own double-spend attempts, which have near-zero impact on block propagation time.

Then, the author postulates that poison transactions will forever remain negligible in practice.

Indeed, a rational but dishonest miner has no reason to abide with the double-spend attempts of other users as there is no profit to be gained. If the miner takes a bribe through a higher transaction fee, then the miner automatically graduates to the position of *organized crime* which, as detailed previously, is an unenviable position if the miner can't exercise power on who gets defrauded.

Moreover, delaying the propagation on a newfound block is an expensive practice, already costing the miner about $15 per second for a blockchain network demonstrating full 1MB blocks. If Bitcoin ever grows to anything significant for the world economy, this degraded latency is at odds with the preservation of the anonymity of the miner through Tor or something similar.

Large blocks are required in practice to give the miners the option to compete on block validation and block propagation. Indeed, with small blocks, this competition still exists, but its impact on the mining market is much smaller. Thus, it is remarkable, paradoxical even, that infrequent blocks - hence bigger blocks for any given transaction rate - timed 10 min apart, appear to be positively contributing, albeit indirectly, to the security of 0-conf transactions. Satoshi Nakamoto appears to have done a wise[3] choice back in 2008 in opting for 10 min as the target duration for blocks.

Competitors[4] who emerged later to compete with Bitcoin appear to have been oblivious to this angle. Even competitive forks which emerged from Bitcoin itself with the intent to forever cap the block size appear to be equally oblivious, as small blocks, which also happen to get full, destroy any hope of ever securing 0-conf transactions, as prioritizing transactions by fees becomes the rational non-fraudulent[5] strategy for miners.

Centralized timestamping authority

The simplest approach to building a pre-consensus signal is probably to build a centralized timestamping authority. Such an authority listens to the Bitcoin network and issues unique high-precision timestamps for any transaction identified by its transaction identifier. An authoritative timestamp includes three values: the transaction identifier, the timestamp and the signature of the authority.

[3] Satoshi refers to "wisdom" in Japanese.
[4] The author is thinking about Litecoin, Ethereum and Ripple in particular.
[5] Yes, the *intent* of fraud matters in the eye of a court of justice. If RBF (replace-by-fee) is acknowledge by participants as being the expected rule of the game, then RBF isn't fraud anymore. To make RBF fraud, all the Bitcoin community has to do is to agree that poison transactions are fraud. Courts aren't behaving like a flawed Ethereum token where code-is-law, despite being broken. Ruling usually makes sense.

In order to make timestamping process scalable, timestamps are only expected to be unique for 1 minute. Once 1 minute has passed, the authority is allowed to re-sign a transaction identifier with a conflicting timestamp. This delay is a tradeoff between performance and transparency. The delay should be long enough to enough that an external observer, located at the edges of the Bitcoin network, can access the integrity of the authority. Yet, in the same time, the delay should be short enough so to keep the internal key-value store within the authority as manageable as possible.

The timestamping process can trivially be sharded through the lowest digits of the timestamps which are used to identify the *shards* rather than the passing of time. If the timestamps are numerically very precise (say 96 bits) then, this sharding trick can be used for any number of shards of practical relevance. Thus, the timestamping problem is an embarrassingly parallel problem, and a centralized timestamping authority can be built at any scale, assuming that someone is willing to pay for the linear hardware costs involved.

Any miner who has access to the centralized signal authority can submit all its transactions to this authority - or conversely rely on the authority to push all the relevant timestamps - and get their corresponding timestamps.

Whenever a conflict is found between transactions by the miner, the miner adopts the transaction associated with earliest timestamp as the correct one, and attempts to propagate this transaction to its peer. As timestamps are guaranteed to be unique, all conflicting transactions are resolved in an unambiguous manner for all miners who abide to the same authority. Those propagating transactions represent the *signal* induced by the authority.

If the cost for the miner to gain access to the centralized authority can be made low enough to recoup the gains that the miner generates from gaining access to a reasonably accurate signal, then it is rational for this miner to adopt this authority as its trusted source of signal.

Naturally, signals compete between themselves, and if the miner can find a *better* signal, then the miner can adopt this alternative signal instead.

If the miner does not have access to any alternative signal, if the authority delivers a reasonably accurate signal - which can be verified by the miner itself by simply monitoring the authority over a long period of time, and finally if the costs involved to gain access to this signal are low enough, then it is rational for the miner to abide to this authority as its signal.

Microlatency payment use case

A centralized timestamping authority can deliver a signal with an operating latency of 1 second to the world at large. Yet, as we will illustrate in this section, there are use cases where latencies as low as 10 ms would be desirable. While such a use case may seem remote to the reader, the author believes that it is of prime relevance because it illustrates the benefits that

can be obtained from a Bitcoin-powered economy where transactions are secured within milliseconds.

Alice is a company that manufactures and sells a variety of home accessories. Alice sells on the web to the world at large, but produces near Shanghai. Bob is a package delivery company who ensures that any parcel it receives can be delivered anywhere in the world. Alice wants to expedite its parcels using Bob. Alice puts a short range RFID on every parcel. The RFID encodes both the intended final destination of the parcel and an URI to submit a Bitcoin payment request for shipment. This URI is a secret intended for Bob only. As gaining access to the secret requires short-range access to the parcel itself, the secret is safe. Alice knows the tariffs of Bob, which are public, but the tariffs depend on the weight of the parcel, and Bob trusts no-one but its own measurements to compute the price for any parcel it receives.

Upon reception of an autonomous truck full of parcels sent by Alice within the secure logistic platform of Bob, a device starts to unload the content of the truck, putting all parcels on a high speed conveyor belt. Every parcel is weighed going through a balance located on the conveyor belt itself. Further down the conveyor belt, a short range RFID reader reads the destination and the secret URI. The IT system of Bob reaches out to the endpoint of Alice, and posts a Bitcoin address as well as the amount of bitcoins to be paid. Alice verifies that the requested payment price originating from Bob reflects her own weight expectation for the parcel. Upon verification, Alice emits a 0-conf transaction to acknowledge her agreement. Bob waits exactly the time it takes to secure the 0-conf transaction to proceed with the next parcel also located on the conveyor belt, mere one meter behind the parcel that has just been paid for. High speed conveyor belts are expensive, and the logistic platform is not a storage facility. Bob cannot afford to have thousands of parcels lingering around while payments are pending. Bob depends on its capacity to secure a payment in 10 ms in order to keep his high speed conveyor belt working a full capacity.

As a final note, the author would like to point out that this use case should not be confused with a *micropayment* use case for Bitcoin. As the time of this writing, shipping any sizeable parcel across the world costs several dollars or more. Those transactions are of non-trivial economic value.

Practical faster-than-light timestamping authority

The minimal theoretical latency for a speed-of-light round trip on earth is 133 ms assuming a distance of 40,000km. Then, in practice, the *ping* that can be reliably achieved from data center to data center is about twice[6] the value of the speed-of-light. At first glance, this would seem to indicate that the minimal theoretical latency of the timestamping authority should be of 266 ms when considering the most distant miner within the network, assuming that the miner and the authority are 40,000 km apart. Even if we tighten this estimate by assuming that Bitcoin mining

[6] A very impressive engineering feat, arguably a “small miracle” of modern communication networks.

will primarily happen in the northern hemisphere[7], it would seemingly appear that claiming that timestamping can happen in less than 100 ms for any miner on earth would violate the speed-of-light constraint.

However, there is a practical way to achieve arbitrarily low latencies from the timestamping authority. As a first requirement, the authority has to distribute itself, spreading its own nodes across the globe in order to reduce its own latency when reaching to its peers. Naturally, this distribution begs the question of the capacity of the authority to never double-sign a given transaction identifier with two conflicting timestamps.

Then, as second requirement, to achieve this seemingly impossible feat, the authority has to establish a convention where the *first byte* of any Bitcoin UTXO outpoint is a geocode that implicitly encodes the desired geo-location where the timestamping is intended to happen. A convention should be established ahead of time within the Bitcoin community to identify the 256 large cities - well distributed across the globe - as geo-targets for the timestamping process.

A payee generating a Bitcoin address embeds its own location into the address by generating a 1-byte vanity address[8] which can be done in a few milliseconds. The generation of the geo-encoded Bitcoin address can even be done ahead of time if needed. The payer is expected to match this address with a payment made from an UTXO entry matching the geocode of the address of the payee.

This scheme can be extended to 2-byte vanity addresses (or more) which would further narrow down the exact intended geo-location of the timestamping process. The fine print will ultimately depend on exact industrial requirements that are not known to the author at the time of this writing.

The Ansible, a peer-to-peer timestamping authority

While a centralized yet distributed timestamping authority can deliver an accurate pre-consensus signal at any operating latency - as illustrated by the previous section - miners could object that the authority itself, being centralized, would introduced a single point of failure into Bitcoin at large.

Thus, we propose to introduce *The Ansible*, a secure peer-to-peer decentralized fault-resilient mechanism to produce an accurate signal. In practice, the Ansible presents itself as an optional networking dialect that can be supported (or not) by any Bitcoin node, much like Compact Block.

[7] The Northern Hemisphere is home to approximately 6.57 billion people which is around 90% of the earth's total human population of 7.3 billion people. https://en.wikipedia.org/wiki/Northern_Hemisphere
[8] The performance numbers given at https://en.bitcoin.it/wiki/Vanitygen indicate that a modern smartphone should be able to generate about 1M addresses per second.

The Ansible is categorized by its participants which include one master and multiple members. The membership to the Ansible is gained by a miner by putting its public key into a block that has reached an age of 10 blocks *plus 60 seconds* within the blockchain. The membership lasts for 100 blocks onward *plus 60 seconds*, unless revoked. This public key ensures that the member can be identified as such.

The “plus 60 seconds” part of the membership rule ensures that change of *master*, as defined in the following, does not collide with the emergence of block propagation races; at least not on a systematic manner.

The rules of the Ansible are as follow:

- Ansible-capable nodes relay all the messages properly signed by non-revoked members of the Ansible.
- At any point in time, any member of the Ansible can cast a public vote of revocation against any of its peers. Unlike other messages, revocation votes are always propagated, but with a limit of 100 distinct messages per member. The revocation is decided based on a majority rule based on the number of members to be found in the blocks aged from 10 to 110 blocks.
- The most aged non-revoked member of the Ansible is automatically elected as the master of the Ansible.
- The master of the Ansible acts as a timestamping authority for the benefit of the other non-revoked members of the Ansible. Members immediately cast a revocation vote if the master fails in its duty of delivering diligent and unique timestamps.
- Members propagate through peer-to-peer the timestamps issued by the master of the Ansible.

The timestamps issued by the master represent the pre-consensus signal issued by the Ansible. Honest miners who are part of the Ansible are *expected*[9] to resolve double-spend attacks according to the pre-consensus signal of the Ansible itself.

The master of the Ansible can make itself seemingly faster-than-light by distributing itself, according to the scheme presented in the previous section.

The Bitcoin security model is built on a small-world network which incentivizes miners to be so densely connected that those miners can effectively be seen one ultra-reliable super-node from the perspective of an external observer. The Ansible *reifies* this perspective through its elective members, but does not interfere with it.

[9] Merely an honest expectation from honest miner to honest miner. Like any signal, miners remain free to abide or not to the signal their are contributing to form with the Ansible.

The hidden Ansible

In order to make themselves invisible to the Bitcoin network at large, the members of the Ansible can leverage a simple mechanism to make themselves known to each other without making themselves known to external observers.

The main benefit from this approach is to make the Ansible more difficult to attack through direct DDOS attacks directly attempted to harm the Bitcoin network.

In order make themselves known to each other, the members propagate in peer-to-peer fashion one message encrypted for every other non-revoked member which contains the URI (uniform resource identifier) where the member itself can be reached within the network. The member can issue a distinct URI for each fellow member in order to provably indoctrinate the member at the origin of the leak if this URI were to be publicly disclosed.

In the case of a distributed setup, each member would issue multiple endpoints matching all the relevant geographical locations, as per the convention established by geo-encoded Bitcoin addresses.

Anticipated results from the Ansible

While the accuracy of the pre-consensus signal which will result from the Ansible can only be known for certain by convincing miners who run an actual implementation of the Ansible, this accuracy can be expected to be very high, probably ranging above 99.99% in practice.

Indeed, a lower accuracy would point out that poison transactions are rampant within Bitcoin at the present time, which does not match the general understanding of the author of the Bitcoin ecosystem. The casual (non-scientific) observations of the author indicate that even considering Bitcoin competitors[10] where users are massively complaining through various online forums about high confirmation times and high fees, the actual double-spend concerns remain remarkably absent.

As any reasonably accurate Ansible triggers a self-fulfilling prophecy of its own making through a virtuous alignment of the miner's interest, the introduction of the Ansible would most likely make 0-conf transactions secure forever in Bitcoin.

Annex: Predicting the winning block with the Ansible

Whenever a race condition on block propagation happens, it is of prime interests to the miners of making an accurate prediction on which block is going to win the block propagation race.

[10] The author thinks of the small-block Segwit fork of Bitcoin for example.

Building a binary classification function capable of predicting which block is going to win the race is fundamentally similar to the problem of predicting which transaction - out of multiple transactions - is going to win the propagation race. This perspective begs the question of using the Ansible, as a timestamping authority, to gain an edge on the block propagation race.

The timestamping process is not processing *transactions* but rather *transaction identifiers* which happen to be sequences of 32 bytes. Those identifiers are not differentiable from the *block identifiers* which as also 32 bytes long. This property, albeit partly accidental, is quite interesting from the perspective of using the Ansible to timestamp the *block hash*. Indeed, as the master of the Ansible cannot differentiate whether it is timestamping a transaction or a block, it cannot specifically interfere with the delivery of a signed timestamp associated with a block hash.

Then, if miners routinely observe that the Ansible happens to be accurate in predicting the outcome from a block propagation race as well, precisely leveraging the timestamp issued for the block hash; then, all miners (except to two having a stake at having their own block winning the race) should rationally take advantage of this information to start working the block that is deemed the winner by the Ansible timestamps.

If miners collectively make the same reasoning, then, orphaned blocks would practically ceased to exist under normal network conditions when the Ansible is working properly.