# Algorand 2021 Performance

By Silvio Micali

Algorand is growing. Today, an average of 500,000 transactions per day are posted on our chain. More than 500 companies are busy developing applications on Algorand, taking advantage of our unique layer-1 smart contracts and the other functionalities that enrich our platform. Their applications will soon generate plenty of new transactions.

This is why, while continuing to add new functionalities to Algorand, we are improving our performance, without sacrificing decentralization, as follows.

## OUR PERFORMANCE MEASURES

1. *Block **proposal** time.* This is the time it takes observers to become aware of which block is a candidate to be permanently added to the chain.

2. *Block **finalization** time*. This is the time needed to ensure that a new block is permanently added to the chain.

3. ***Finalized** transactions per second* (TPS).

## OUR 2021 PERFORMANCE

- *Block **proposal** time will remain 0.5 seconds.*
  (Even though our block size will grow from 5,000 to 25,000 transactions.)

- *Block **finalization** time will shrink from 4.5 to 2.5 seconds.*

- *Our **finalized** TPS will grow from 1,000 to 46,000*.
  (Thanks to a truthful approach to block pipelining.)

## OUR PRINCIPLED EVOLUTION

Algorand's overarching goal is providing a truly decentralized, public, permissionless network that scales perfectly and eliminates the performance drawbacks of first-generation blockchains. Decentralization and security are fundamental principles in Algorand. They will be respected by this performance improvement and all future enrichments of our platform.

## 1. Finality Metric and Finalized Performance

WHICH METRIC?

Increasingly, “performance*”* is a word used loosely and liberally across the blockchain universe. Of course, speed matters. But: *speed of what?*

Consider the following performance claim:

> *“A 10K-transaction block is proposed every 0.5 seconds.”*

Two main questions arise:

1. *Does the claim imply a latency of half a second?*
2. *Does the claim imply a throughput of 20K TPS?*

The answer to the first question is NO. Block proposition is just the first step of a journey with no guarantees of safe arrival. Stating only block-proposal speed conveniently ignores the time needed to determine the block’s finality, assuming it will be finalized!

The answer to the second question is also NO. By itself, block-proposal speed provides no throughput guarantees, because it conveniently ignores the case in which a proposed block $B$ is *not* finalized. In this case, not only the transactions of $B$ will have to be processed again, but also those of the blocks $B+1, \dots, B+k$, if $k$ blocks were proposed during the failed finalization of $B$. Indeed, the transactions of these $k$ blocks depend on the validity of $B$’s transactions. Thus, whenever the finalization of $B$ fails, there have been 0 TPS for the entire duration of $B$’s finalization.

In sum, the above performance claim is based on a very sketchy metric.

THE FINALITY METRIC

The very goal of a blockchain is to provide a *complete* and *immutable* sequence of transactions.

Quick block proposition is the equivalent of being quickly told that “your money is on the way.” Finality is equivalent to your having “money in your hand.” But then, it is up to you:

- If you prefer to ship your goods, release your digital art, issue your insurance, etc. when you are just told that your money is on the way, then focus on block proposition.

- If you prefer to ship your goods, etc. when you have actually cashed your payment, then focus on block finality.

For us at Algorand, finality is what really matters and what we deliver.

OUR FINALIZED PERFORMANCE

Sticking to the finality metric, our 2021 performance will be as follows:

- ***Finalized*** *Latency:* $\approx$ 2.5 seconds.
- ***Finalized*** *Throughput:* $\approx$ 46,000 TPS.

## 2. Our New Finalized Latency

An Algorand block is a large object. Today it comprises up to 5,000 transactions, and in 2021 it will comprise up to 25,000 transactions. Propagating a large object takes time.

To speed up the processes of proposing and finalizing a new block $B$, we have developed a more compact way to specify it.

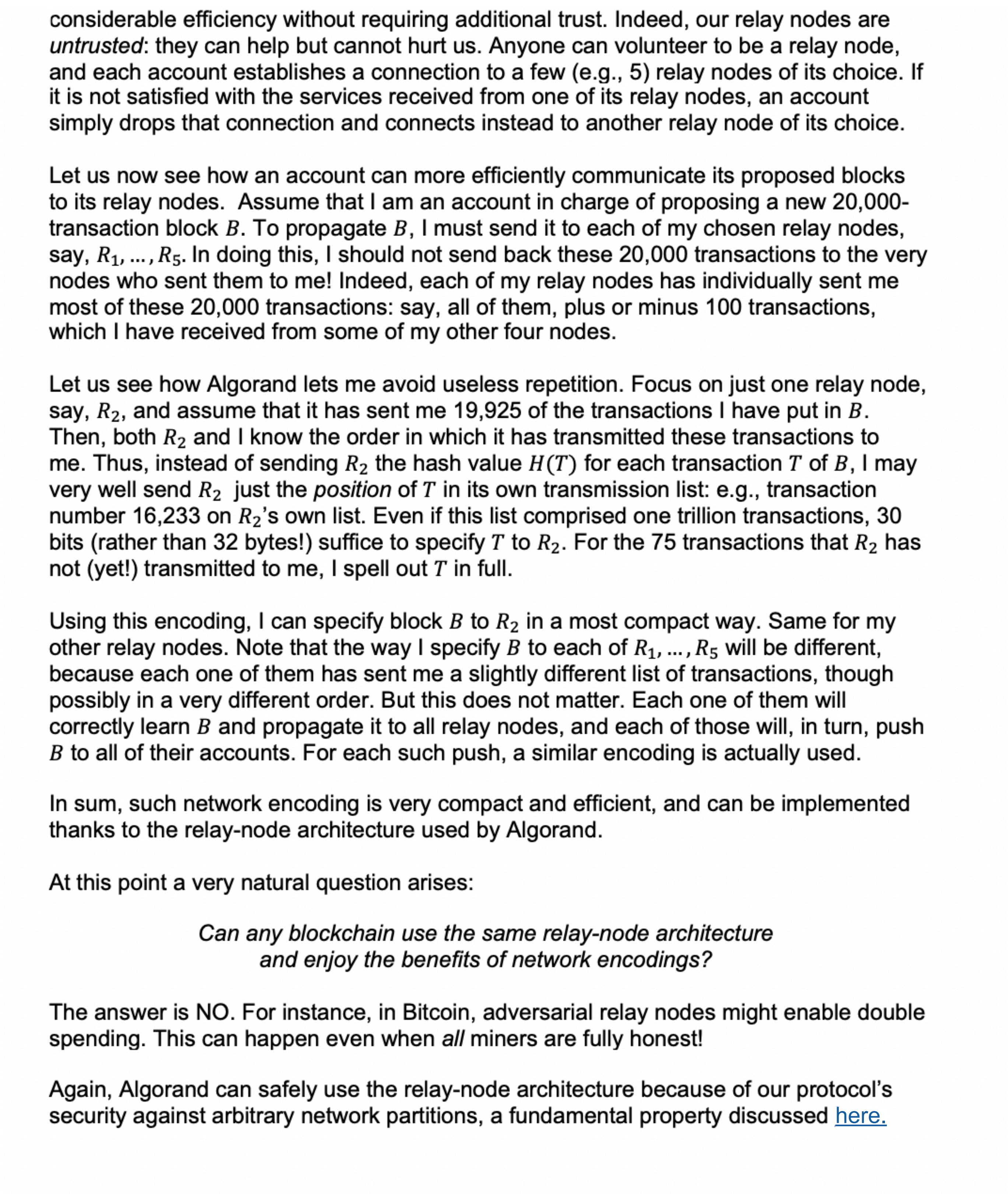
NETWORK CODING

Once we add a block $B$ to the immutable record provided by the Algorand blockchain, $B$'s transactions must be explicitly spelled out. Indeed, such a block $B$ may be consulted years later, by people who do not have any knowledge of our shared context.

But during the generation of $B$, we can and indeed do come up with shorter `names' for its transactions, taking advantage of the fact that, since transactions are propagated throughout the network, at any point in time any two of us have seen the same transactions, plus or minus just a few of them. For instance, for most transactions $T$, to inform you that my proposed block includes $T$, rather than spelling out $T$ in its entirety, I may very well send you just the 32-byte hash value $H(T)$. In fact, practically speaking, no two transactions have the same hash value.

To improve our performance, we will further reduce these 32 bytes to just a few *bits*, by leveraging the very specific ways Algorand communication network operates.

In Algorand, accounts do not exchange messages directly with one another via peer-to-peer gossiping. Accounts send the messages they want to propagate to their relay nodes. Relay nodes propagate messages to each other and push them to those accounts with whom they have a connection. Thanks to a deep property of our protocol, *security against arbitrary network partitions*, Algorand's architecture introduces



considerable efficiency without requiring additional trust. Indeed, our relay nodes are *untrusted*: they can help but cannot hurt us. Anyone can volunteer to be a relay node, and each account establishes a connection to a few (e.g., 5) relay nodes of its choice. If it is not satisfied with the services received from one of its relay nodes, an account simply drops that connection and connects instead to another relay node of its choice.

Let us now see how an account can more efficiently communicate its proposed blocks to its relay nodes. Assume that I am an account in charge of proposing a new 20,000-transaction block $B$. To propagate $B$, I must send it to each of my chosen relay nodes, say, $R_1, \dots, R_5$. In doing this, I should not send back these 20,000 transactions to the very nodes who sent them to me! Indeed, each of my relay nodes has individually sent me most of these 20,000 transactions: say, all of them, plus or minus 100 transactions, which I have received from some of my other four nodes.

Let us see how Algorand lets me avoid useless repetition. Focus on just one relay node, say, $R_2$, and assume that it has sent me 19,925 of the transactions I have put in $B$. Then, both $R_2$ and I know the order in which it has transmitted these transactions to me. Thus, instead of sending $R_2$ the hash value $H(T)$ for each transaction $T$ of $B$, I may very well send $R_2$ just the *position* of $T$ in its own transmission list: e.g., transaction number 16,233 on $R_2$'s own list. Even if this list comprised one trillion transactions, 30 bits (rather than 32 bytes!) suffice to specify $T$ to $R_2$. For the 75 transactions that $R_2$ has not (yet!) transmitted to me, I spell out $T$ in full.

Using this encoding, I can specify block $B$ to $R_2$ in a most compact way. Same for my other relay nodes. Note that the way I specify $B$ to each of $R_1, \dots, R_5$ will be different, because each one of them has sent me a slightly different list of transactions, though possibly in a very different order. But this does not matter. Each one of them will correctly learn $B$ and propagate it to all relay nodes, and each of those will, in turn, push $B$ to all of their accounts. For each such push, a similar encoding is actually used.

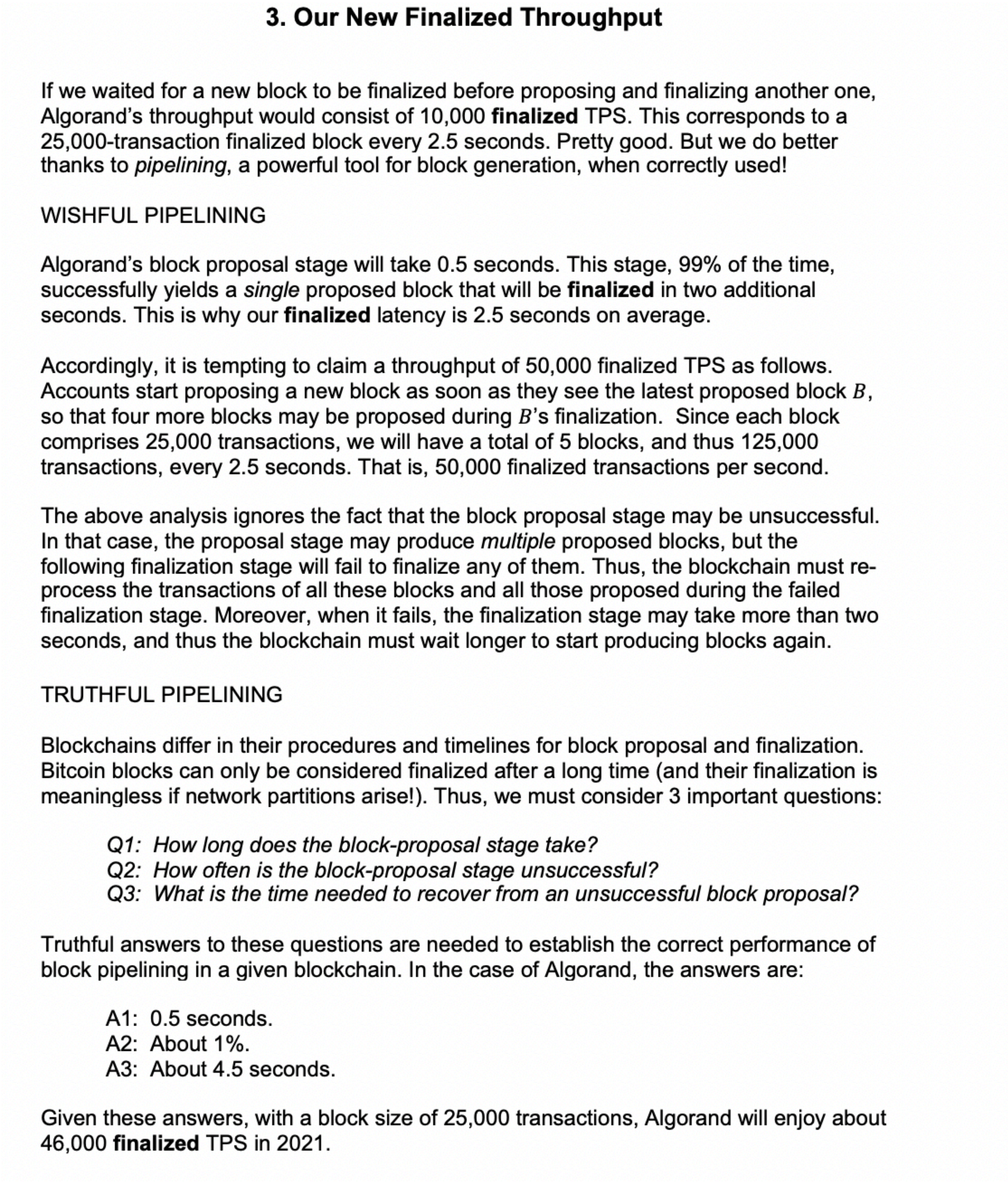
In sum, such network encoding is very compact and efficient, and can be implemented thanks to the relay-node architecture used by Algorand.

At this point a very natural question arises:

*Can any blockchain use the same relay-node architecture and enjoy the benefits of network encodings?*

The answer is NO. For instance, in Bitcoin, adversarial relay nodes might enable double spending. This can happen even when *all* miners are fully honest!

Again, Algorand can safely use the relay-node architecture because of our protocol's security against arbitrary network partitions, a fundamental property discussed here.



## 3. Our New Finalized Throughput

If we waited for a new block to be finalized before proposing and finalizing another one, Algorand's throughput would consist of 10,000 **finalized** TPS. This corresponds to a 25,000-transaction finalized block every 2.5 seconds. Pretty good. But we do better thanks to *pipelining*, a powerful tool for block generation, when correctly used!

WISHFUL PIPELINING

Algorand's block proposal stage will take 0.5 seconds. This stage, 99% of the time, successfully yields a *single* proposed block that will be **finalized** in two additional seconds. This is why our **finalized** latency is 2.5 seconds on average.

Accordingly, it is tempting to claim a throughput of 50,000 finalized TPS as follows. Accounts start proposing a new block as soon as they see the latest proposed block $B$, so that four more blocks may be proposed during $B$'s finalization. Since each block comprises 25,000 transactions, we will have a total of 5 blocks, and thus 125,000 transactions, every 2.5 seconds. That is, 50,000 finalized transactions per second.

The above analysis ignores the fact that the block proposal stage may be unsuccessful. In that case, the proposal stage may produce *multiple* proposed blocks, but the following finalization stage will fail to finalize any of them. Thus, the blockchain must re-process the transactions of all these blocks and all those proposed during the failed finalization stage. Moreover, when it fails, the finalization stage may take more than two seconds, and thus the blockchain must wait longer to start producing blocks again.

TRUTHFUL PIPELINING

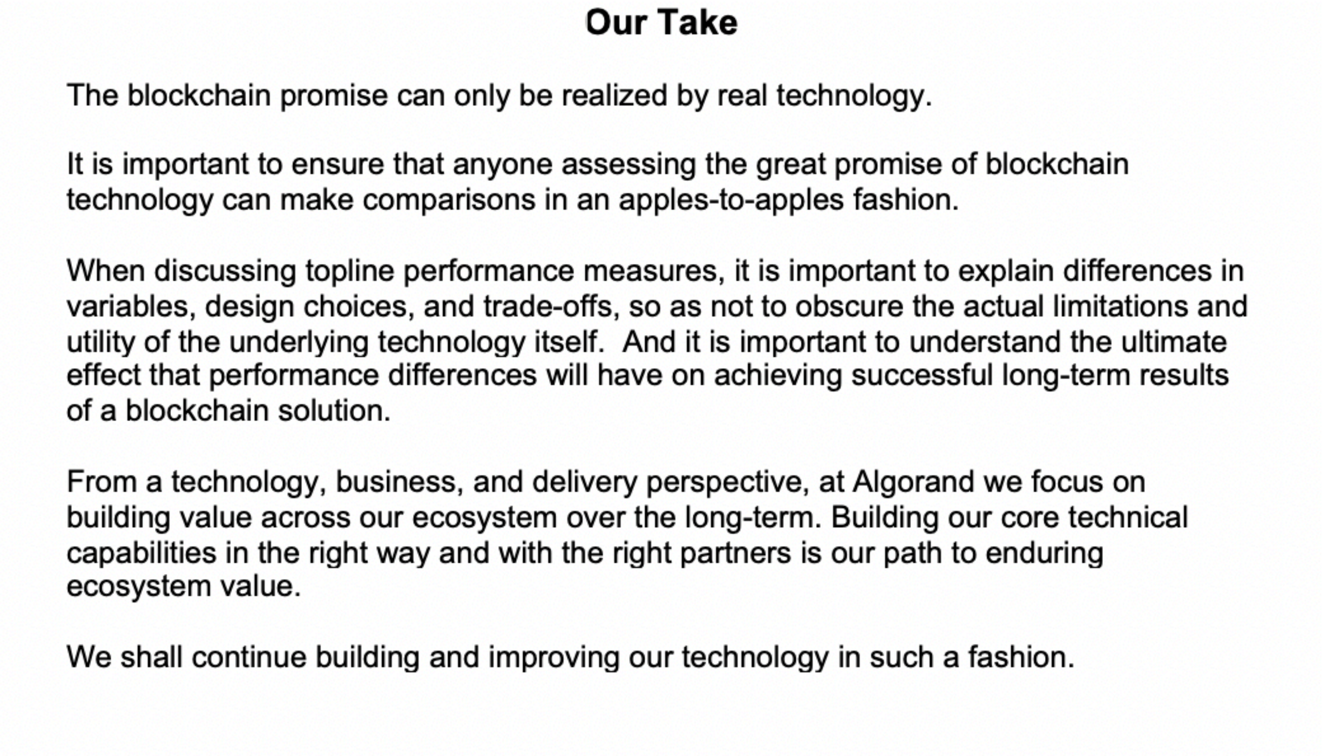
Blockchains differ in their procedures and timelines for block proposal and finalization. Bitcoin blocks can only be considered finalized after a long time (and their finalization is meaningless if network partitions arise!). Thus, we must consider 3 important questions:

*Q1: How long does the block-proposal stage take?*
*Q2: How often is the block-proposal stage unsuccessful?*
*Q3: What is the time needed to recover from an unsuccessful block proposal?*

Truthful answers to these questions are needed to establish the correct performance of block pipelining in a given blockchain. In the case of Algorand, the answers are:

A1: 0.5 seconds.
A2: About 1%.
A3: About 4.5 seconds.

Given these answers, with a block size of 25,000 transactions, Algorand will enjoy about 46,000 **finalized** TPS in 2021.



## Our Take

The blockchain promise can only be realized by real technology.

It is important to ensure that anyone assessing the great promise of blockchain technology can make comparisons in an apples-to-apples fashion.

When discussing topline performance measures, it is important to explain differences in variables, design choices, and trade-offs, so as not to obscure the actual limitations and utility of the underlying technology itself. And it is important to understand the ultimate effect that performance differences will have on achieving successful long-term results of a blockchain solution.

From a technology, business, and delivery perspective, at Algorand we focus on building value across our ecosystem over the long-term. Building our core technical capabilities in the right way and with the right partners is our path to enduring ecosystem value.

We shall continue building and improving our technology in such a fashion.

**SILVIO MICALI** | Founder, Algorand

Silvio Micali has been on the faculty at MIT, Electrical Engineering and Computer Science Department, since 1983. Silvio's research interests are cryptography, zero knowledge, pseudorandom generation, secure protocols, and mechanism design and blockchain. In particular, Silvio is the co-inventor of probabilistic encryption, Zero-Knowledge Proofs, Verifiable Random Functions and many of the protocols that are the foundations of modern cryptography.

In 2017, Silvio founded Algorand, a fully decentralized, secure, and scalable blockchain which provides a common platform for building products and services for a borderless economy. At Algorand, Silvio oversees all research, including theory, security and crypto finance.

Silvio is the recipient of the Turing Award (in computer science), of the Gödel Prize (in theoretical computer science) and the RSA prize (in cryptography). He is a member of the National Academy of Sciences, the National Academy of Engineering, the American Academy of Arts and Sciences and Accademia dei Lincei.

Silvio has received his Laurea in Mathematics from the University of Rome, and his PhD in Computer Science from the University of California at Berkeley.