Hyperspace: Fast, Cheap, and Private Storage

Mark Huetsch
mark@hspace.app

Wang Chao
wangchao@hspace.app

Version 1.0, December 6, 2018

1 Introduction

Hyperspace is a blockchain-based storage marketplace featuring its own digital currency, Space Cash (XSC). Hyperspace facilitates permissionless and trustless use of cloud object storage. Data is end-to-end encrypted for complete privacy and Reed-Solomon encoding is used so that storage is redundant and resistant to data loss. Storage operations can happen concurrently with multiple hosts for higher transfer speed. The blockchain is secured with Proof of Work and a hybrid consensus model with additional layers of transaction security is being developed. The two primary participants in the marketplace, storage renters and hosts, use Space Cash in storage contracts. Renters pay for storage use and hosts post collateral to guarantee their service. The marketplace encourages competition, incentivises quality-of-service, and drives commodification of data storage services so that the prices tend toward the marginal cost of production.

2 Ease of Use

Many attempts at blockchain-based distributed storage have been made, including Sia, the codebase upon which Hyperspace is originally based, but none of these attempts have gained significant traction, partially due to difficulty of use and lack of proper economic incentives.[1] Hyperspace aims to tackle both of these issues - with usability being the foremost concern.

2.1 Lightweight Nodes

To rent or host storage, users install client software that represents a node in the Hyperspace network. Full nodes maintain a local copy of the entire blockchain and record new transactions such as rental contracts and transfer of funds. However, full nodes have gained a deserved reputation of being unusable for average users. Lightweight nodes, also called light nodes, do not require a local copy of the entire blockchain but instead use a protocol called Simplified Payment Verification (SPV) to determine which blocks they need and request them from other full nodes. The first generation of SPV relied on Bloom Filters, which leaked client information and significantly increased computing resource demands on full nodes.[2] Hyperspace uses Goulomb coded sets (GCS) to generate efficient filters which are included with block headers and broadcasted to all nodes.[3] Lightweight nodes are essential to widespread user adoption and can be implemented in many different ways, such as desktop software, mobile app, or web service.

2.2 3rd-Party Contract Management

Hyperspace data storage operates via a narrowly defined smart contract. Nodes announce that they would like to serve as hosts by broadcasting a special host announcement transaction. Renters keep track of these hosts and, using a ranking database, choose to form contracts with a desired number of hosts. Renters negotiate with each host directly to form a contract for a fixed amount of Space Cash and for a fixed period of time. The host puts in collateral and the renter pre-pays funds that it is willing to spend. Both parties sign and then one of them broadcasts the transaction to the network.

When data is transferred between renter and host, the renter and host open a point-to-point payment channel whereby they each sign a contract revision, transferring funds as data is being transferred. As each chunk of data is transferred, the renter uses a locally stored merkle root to verify that the data received is indeed the data requested. Once data is done being transferred, either party can broadcast the revised contract to the blockchain via a transaction.

Two problems occur with direct user management of contracts. First, when a contract expires, after an agreed time of typically 3 to 6 months, the user must be online to pay for and sign new contracts with hosts, otherwise the user will lose data. Second, many users may find it inconvenient to acquire Space Cash to pay hosts, and may prefer paying with their local fiat currency.

2.2.1 User Experience and Fiat Payments

Hyperspace is being designed so a 3rd party (contractor) can take on responsibility for storage contract management. The contractor can offer value-added services, including their own user interface, fiat payment options or storage entitlements, and manage contracts with hosts on the renter’s behalf. The protocol allows contract revision and signing to be delegated to a contractor server, while the actual storage operations remain directly between the storage renter and the hosts. For example, when a user wants to upload a file, he can request the contract data from the contracting server and then open connections with the hosts. As each chunk of data is transferred between renters and hosts, the renter can indicate to the contracting server that it should sign a contract revision in its payment channel with the host. If the renter stops telling the contractor to sign, the host will stop sending data, broadcast the most recent contract revision to the blockchain, and the upload will finish. A similar process can occur with downloads. In this manner, cryptocurrency payment can be offloaded to a 3rd party while the vast majority bandwidth usage still occurs directly between renter and host. This resolves a key design problem of otherwise requiring all files to be first copied to a 3rd party node and then forwarded to the different storage hosts or vice-versa.

2.2.2 Unattended Contract Renewal and Data Repair

A similar process allows contracts to be renewed and data to be repaired without the renter needing to be online. The contractor can continuously monitor the health of its customers' contracts. When a host goes

down or loses data or when a contract expires, the contractor can form new contracts with new hosts and identify the hosts holding redundant copies of the missing chunks of data. The contractor can then communicate both with the old redundant host and the newly contracted host and tell the host with the data that needs to be copied to upload the data to the new host. The process here is much like the new host is acting as the original user when the user wants to download: when repairing files the contracting server signs the user's contracts for him. The old host simply sends the data to a new host instead of the end user. Simultaneously, the new host is paid to download the redundant data via the newly formed contract that is also being signed by the 3rd party contractor. In this way, a reliable 3rd party contracting service can ensure that a user will not lose data even if the user is not online to repair data or renew contracts. Simultaneously, most of the bandwidth usage in a repair or renewal occurs between the old host and the new hosts. The 3rd party does not use much bandwidth relative to the amount of data being transmitted between hosts.

2.3 File Recovery from Seed

To use rented storage on the Hyperspace network, users currently setup a dedicated local folder on their device, and all files in that folder will be uploaded with redundancy to hosts on the Hyperspace network. However, to download or restore files that the user may have lost or accidently deleted from his device, locally stored metadata is used. Should the user have lost his device with the local metadata or the local hard drive have failed, access to the user’s files would be lost. Hyperspace is being designed to overcome these constraints and offer complete distributed object storage that does not rely on local metadata for download or streaming data from the network. This would likely involve hosts storing the encrypted metadata necessary to retrieve full files. Information about which hosts hold the metadata for a given public key attached to a seed could either be broadcast to the blockchain via a transaction or queried via a flood request of existing full nodes or hosts. The former approach would be more reliable but would also consume valuable blockchain space. This is an area still under research.

3 Scalability and Privacy

3.1 The Problem of Blockchain Growth

Blockchains suffer from a number of problems. Chief among them is that, currently, blockchains only grow and never shrink. Each transaction on the blockchain is permanent. This means that after a few years it can become very expensive and slow to set up a full node.[4] In the original Bitcoin whitepaper Satoshi Nakamoto speculated that blockchains can be safely pruned of old, used transactions but thus far no widely used consensus database has used such a mechanism.[5] Therefore an important objective in the short-term is to slow the rate at which a blockchain grows. This means data should be kept off the blockchain unless it is absolutely necessary.

3.2 Schnorr Signature Aggregation

One useful tactic Hyperspace applies is Schnorr signature aggregation. Traditionally, when a transaction is built, each input to the transaction is signed separately. Signatures take up a fair amount of space, so a nice optimization is to aggregate all input signatures into one signature per transaction using a technique called MuSig.[6] Verification of one signature per transaction instead of many should also lead to a faster blockchain sync speed. Blockchain size growth can be further brought down if nodes are willing to collaborate with each other when creating transactions. To do so, they can use an established technique called CoinJoin, a technique where multiple inputs and outputs from different parties are merged - effectively merging multiple transactions into one.[7] Combining Schnorr signature aggregation with CoinJoin has the potential for significant blockchain space savings. This technique would have the possible side effect of partially anonymizing the source of funds to recipient outputs as a set of different owners' inputs could be aggregated to a set of different recipients' outputs. Such transaction privacy has both potentially positive and negative consequences, but the cost of not using such a CoinJoin technique is blockchain bloat.

3.3 Contract Signature Aggregation

This signature aggregation technique can be applied in numerous applications. For instance, when renters and hosts sign file contracts, they can aggregate their signatures, thus reducing the signature data cost of a file contract on the blockchain by 50%. This again has the side effect of anonymizing a contract from its host. Hosts publish their public key when they announce on the blockchain that they are open for hosting service. Using Schnorr signature aggregation for file contracts means that an aggregate public key of both the renter and the host is published with each file contract and thus a file contract is not publicly linkable to a given host.

3.4 Multisig Support and Atomic Swaps

Schnorr signature aggregation also means that multisig addresses are both invisible - not publicly recognizable as a multisig address - and efficient. They are efficient in the sense that when one spends from a multisig address using an aggregate signature, one aggregate public key and signature needs to be published in the spending transaction. A related set of techniques called adaptor signatures and scriptless scripts allows us to perform cross-chain atomic swaps that should be indistinguishable from normal transactions to non-swap participants.[8,9] There is further potential scripting logic that could be invisibly and efficiently processed using scriptless scripts and this is an open area of research.

4. Hypernodes

A key problem facing many cryptocurrencies is that the typical user wants to use a light node, but light nodes need a large and reliable network of full nodes to be secure. However, for many currencies, including Bitcoin, there is not a strong short-term incentive to run a full node. Rather, there are many disincentives, including hardware and bandwidth cost. Hyperspace is being designed to incentivize certain full nodes, called Hypernodes, to share in the block reward with miners in exchange for agreed

quality-of-service. Hypernodes can also represent additional layers of transaction security in a hybrid consensus model.

4.1 Significant Stake and Proof of Burn

An important part of being a useful full node is having dedicated equipment and reliable connectivity for robust uptime. Consequently, a Hypernode operator should indicate that they have a locked stake in the network's currency, Space Cash (XSC), and they should further demonstrate their commitment by burning some of that stake upon Hypernode activation. The Hypernode can demonstrate such by transmitting a certain amount of XSC to an address under their control and transmitting another, smaller amount of XSC to a null, permanently unspendable address - effectively "burning" the latter amount of coins. Only nodes that have demonstrated that they have locked a certain amount of coins and burned another amount will be eligible to receive Hypernode awards. Furthermore, the network will periodically query Hypernodes and if the Hypernode is unresponsive it will lose its status of eligibility for block rewards until it relocks and reburns more XSC. Proper cost values for Hypernode stake and burn are tricky to decide as they put a hard upward bound on the number of Hypernodes that can be active on the network at any given time. Such values may need adjustment in the future, but initially a cost of 1,000,000 XSC is deemed reasonable, with 800,000 XSC being locked and 200,000 XSC being burned upon activation. The activation cost can decrease with each block until it is at a low, fixed amount so that as the network matures the network can support a larger number of Hypernodes.

4.2 Multiple Layers of Transaction Security through a Hybrid Consensus Model

Proof-of-Work-only consensus mechanisms have seen a number of security vulnerabilities in the wild, including 51% attacks, mining pool centralization, and miners generating empty blocks. With regards to 51% attacks causing chain reorganizations in particular, Hyperspace currently follows Bitcoin's lead and partially mitigates this via a 10-minute block time. This long block time, however, is unfriendly to users. Hyperspace is being designed with a Decred-inspired hybrid approach whereby Hypernodes compete to earn block rewards by consuming computational resources beyond electricity in order to vote on a previous block's validity.[10] A randomly selected quorum of such Hypernodes would significantly reduce potential attacks by malicious miners or pools as an attacker would need to control the quorum as well as the mining hardware. Such added resilience to attacks will allow to significantly reduce the network's block time to a more user-friendly number, such as 1 block per minute. Furthermore, such a quorum could vote to reject even valid blocks from a miner who is submitting empty blocks when there are transactions in the transaction pool. Below are outlined secondary, tertiary, and potentially even quaternary sources of security, namely Proof of Bandwidth, Proof of Capacity, and Proof of Stake. Hypernodes could participate in Proof of Bandwidth and Proof of Capacity.

4.3 Proof of Bandwidth

Proof of Bandwidth is predicated on the idea that nodes compete to solve a puzzle by burning bandwidth instead of burning electricity. It is, in essence, another type of Proof of Work, using a different mechanism

than the traditional approach espoused in Bitcoin. The process is described as follows. First, as a Hypernode, choose a random but ordered set of 4 nodes based on the set of active Hypernodes, deterministically randomized via the previous block id and a nonce. Then, hash the concatenation of the previous block id with your public key, which was broadcast in the Hypernode collateral announcement. Take this message, forward it to the next node in the list. That node appends its public key to the message, hashes the concatenation again, and forwards the result to the next node in the list. Once the final node has been reached, that node should take the message, sign it, and then concatenate its public key and signature. It should then pass the result to the previous node, which should do the same. The process should be repeated all the way back to the original. The initial node, now that it has the list of public keys, should then verify the passed message and verify each signature. If valid, the initial node should append the previous block id and the nonce used to generate the ordered list of nodes. This is a completed bandwidth proof.[11]

Given a finished bandwidth proof, a node should make a hash of the proof. If the result is less than a certain number, called the bandwidth target, then the node can submit the proof as a bandwidth ticket purchase to the bandwidth ticket pool. The bandwidth target should adjust dynamically based on how many bandwidth proofs are in the bandwidth pool. When a proof of work miner is preparing a new block, the miner must include votes about the validity of the previous block from 5 bandwidth ticket participants. Given that each bandwidth proof contains 5 participants, there are a total of 25 possible voters and there must be at least 13 yes votes for the given block to proceed. If there are a majority of no votes, the transactions and block reward from the previous block are invalidated. If less than 13 yes votes and less than 13 no votes are given, then the previous block will be orphaned as nothing will be able to extend it. Once a block is validated and matured, the bandwidth component of the block reward will be evenly distributed to all those who evenly cast a vote and the Proof of Work reward will be distributed to the miner.

4.4 Proof of Capacity

A near parallel to the bandwidth proof voting mechanism can be replicated using Proof of Capacity instead. Various mechanisms on how to to do this have been implemented and researched in other cryptocurrencies.[12,13,14] This can provide a tertiary layer of consensus security, allowing Hyperspace to further drop the blocktime.

4.5 Proof of Stake

For users who wish to participate in network validation but do not have the desire to contribute bandwidth or hard drive space as a Hypernode, Proof of Stake offers a well-tested alternative. Instead of doing work by “spending” bandwidth or storage capacity, a user can spend tokens directly by locking them up for a period of time in order to purchase a stake ticket. The stake ticket then enters a ticket pool and the validation mechanism works just like the bandwidth and capacity ticket processes.

5. Token Economics

Token economies are based on “tokenized” goods and services, physical or digital, that are limited in supply and useful to a broader audience. Network participants earn, spend or stake tokens on contracts, exchanging a commodity and earning rewards for related services.

On the Hyperspace marketplace, digital storage space is the commodity, and services to store data, secure transactions or manage contracts are being provided by and to the users of the network. In strict technical terms Space Cash (XSC) is a coin not a token, but its economic characteristic compares to that of a utility token. Using Space Cash (XSC) instead of “real” money allows to significantly reduce costs and speed of transactions, overcome barriers of different country-specific currencies and eliminates the need for intermediaries such as banks and payment services as part of the peer-to-peer transactions. In short - it removes friction. Theoretically, Hyperspace could have been using other token protocols such as ERC20 but the generic use of its own coin Space Cash (XSC) built into the core protocol has significant advantages in regards to simplicity, performance, scalability, and resilience.

Unlike Bitcoin, Space Cash (XSC) coin supply is not limited, since digital storage space in general and consequently capacity offered on the Hyperspace network in particular is expected to keep growing exponentially. Moreover, a consistent source of newly generated coins ensures a steady incentive for miners to validate transactions beyond the marketplace for transaction fees. Near-term coin inflation during the initial ramp-up of the network is being mitigated with a decreasing block reward schedule, requirement to lock collateral in each contract for the duration and fixed coin “burns” for Hypernode activation. Eventually the supply emission per year will become fixed. This will occur when the annual inflation rate is about 2%.

While the utility of Space Cash (XSC) is currently confined to the Hyperspace storage marketplace, the support of atomic swaps will allow integration with other blockchain-based marketplaces in the foreseeable future.

6. Licensing

The code for running a Hyperspace node is open source and published under an MIT license.[15] Hyperspace uses open APIs in order encourage both proprietary and open source 3rd party implementations of storage interfaces, wallets, and other services that utilize the storage network.

References

1. D. Vorick, L. Champine, “Sia: Simple Decentralized Storage” https://sia.tech/sia.pdf, November 2014.
2. M. Hearn, M. Corallo, “Connection Bloom filtering”, https://github.com/bitcoin/bips/blob/master/bip-0037.mediawiki, October 24, 2012.
3. O. Osuntokun, A. Akselrod, “Compact Block Filters for Light Clients”, https://github.com/bitcoin/bips/blob/master/bip-0158.mediawiki, May 24, 2017.
4. J. Lopp, “Could SPV Support a Billion Bitcoin Users? Sizing up a Scaling Claim” https://www.coindesk.com/spv-support-billion-bitcoin-users-sizing-scaling-claim, July 30, 2017.
5. S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System; Part 7. Reclaiming Disk Space”, https://bitcoin.org/bitcoin.pdf, October 31, 2008.
6. G. Maxwell, A. Poelstra, Y. Seurin, and P. Wuille, “Simple Schnorr Multi-Signatures with Applications to Bitcoin”, https://eprint.iacr.org/2018/068.pdf, May 20, 2018.
7. G.Maxwell, “CoinJoin: Bitcoin privacy for the real world”, https://bitcointalk.org/?topic=279249, August 22, 2013.
8. A. Poelstra, “Scriptless Scripts”, https://download.wpsoftware.net/bitcoin/wizardry/mw-slides/2017-05-milan-meetup/slides.pdf, May 10, 2017.
9. M. Huetsch, “Hyperspace-compatible cross-chain atomic swapping”, https://github.com/HyperspaceApp/atomicswap, September 15, 2018.
10. The Decred Developers, “Decred Documentation: Hybrid Design”, https://docs.decred.org/research/hybrid-design.
11. M. Ghosh, M. Richardson, B. Ford, and R. Jansen, “A TorPath to TorCoin: Proof-of-Bandwidth Altcoins for Compensating Relays”, https://www.nrl.navy.mil/itd/chacs/sites/www.nrl.navy.mil.itd.chacs/files/pdfs/14-1231-1559.pdf, 2014.
12. S. Gauld, F. Ancoina, R. Stadler, “The Burst Dymaxion: An Arbitrary Scalable, Energy Efficient and Anonymous Transaction Network Based on Colored Tangles”, https://www.burst-coin.org/wp-content/uploads/2017/07/The-Burst-Dymaxion-1.00.pdf, December 27, 2017.
13. S. Park, A. Kwon, G. Fuchsbauer, P. Gazi, J. Alwen, and K. Pietrzak, “SpaceMint: A Cryptocurrency Based on Proofs of Space”, https://eprint.iacr.org/2015/528.pdf, 2015.
14. H. Abusalah, J. Alwen, B. Cohen, D. Khilko, K. Pietrzak, and L. Reyzin, “Beyond Hellman’s Time-Memory Trade-Offs with Applications to Proofs of Space”, https://eprint.iacr.org/2017/893.pdf, 2017.
15. “MIT License”, https://en.wikipedia.org/wiki/MIT_License.