February 18, 2022

Galaxy Digital Research

The Future of Bitcoin Mining Protocols: Making Every Watt Count

Rachel Rybarczyk, VP Galaxy Digital Mining
rachel.rybarczyk@galaxydigital.io | @_rrybarczyk

In this report, we explain how Bitcoin miners communicate with pools, provide an overview of the existing protocol (Stratum V1), and discuss the improvements offered by Stratum V2 and paths to its adoption by Bitcoin network participants.

Executive Summary

When Bitcoin miners work with mining pools, as the vast majority of them do, they communicate with the pools using a messaging protocol called Stratum V1, which organizes the creation of blocks and the submission of hashes by the miners. As the years have progressed and Bitcoin has grown in popularity, so has network hashrate. Stratum V1 has served the community well, but the time has come for a protocol that is robust enough to handle today's high hashrate. Stratum V2 is a highly anticipated new Bitcoin mining protocol that is poised as the next generation of the Stratum V1 protocol and brings many improvements for both miners and pools alike. Although the Stratum V2 protocol is big news, the project has not garnered the attention it deserves. This blog aims to change that by providing an introduction to Stratum V2; first, the rationale for why a new protocol is established, along with some desirable characteristics of a robust mining protocol. In this report, we also give an overview of the existing Stratum V1 protocol, explain the improvements offered by Stratum V2, and then discuss efforts to promote adoption of the new protocol.

Introduction

Examining network hashrate is a good place to start to contextualize Bitcoin mining firmware, a topic that may be new to some readers. Network hashrate is a measure of compute power on the Bitcoin network, and computing power is contributed by Bitcoin ASIC machines, commonly referred to as ASICs or mining devices. Figure 1 shows the network hashrate of all time on a log scale. The general trend is that hashrate increases overtime.

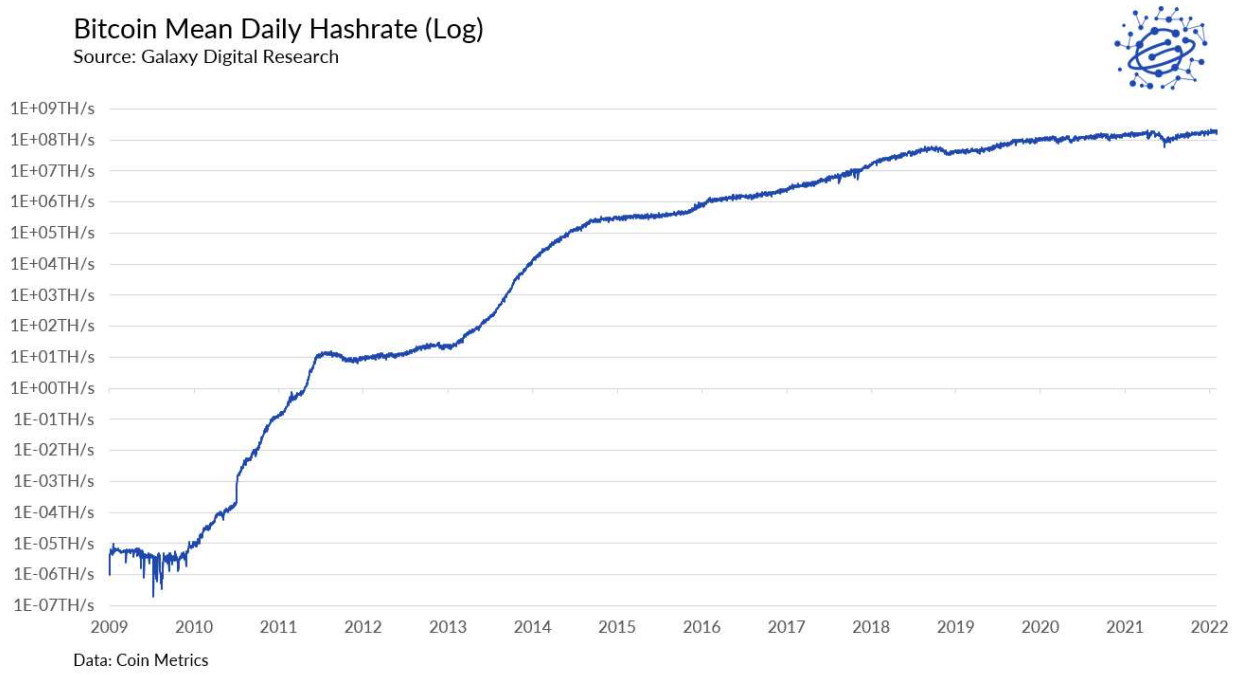

Figure 1: Bitcoin Mean Daily Hashrate (Log)

Regardless of the type of machine connected to the Bitcoin protocol (CPU, GPU, or ASICs), they connect to the network through a mining protocol, most commonly implemented in the machine's firmware. To date, there have been four main mining protocols: the original Bitcoin client (CPU mining), getwork and getblocktemplate (gbt) (which were used with FPGA and GPU mining), and Stratum (the protocol used in ASIC mining).

Stratum has traditionally only had one version of the protocol, most commonly referred to as Stratum V1. Currently, multiple efforts are underway to work on its successor, Stratum V2. In this blog, we will review the current protocol, its capabilities and limitations and also compare that to Stratum V2.

Industry Conditions Leading to Stratum

To understand why a new mining protocol is in desperate need, it helps to look at the historical conditions that called for a change in past mining protocols. Figure 2 shows the network hashrate of all time and illustrates how the space has transitioned between mining protocols over the years. This figure illustrates that new mining protocols naturally emerge when the network hashrate makes a *major* and *permanent* jump. And, with the advent of new and more powerful hashing machines comes the need for a protocol that is designed to handle the new load in an efficient and secure manner.

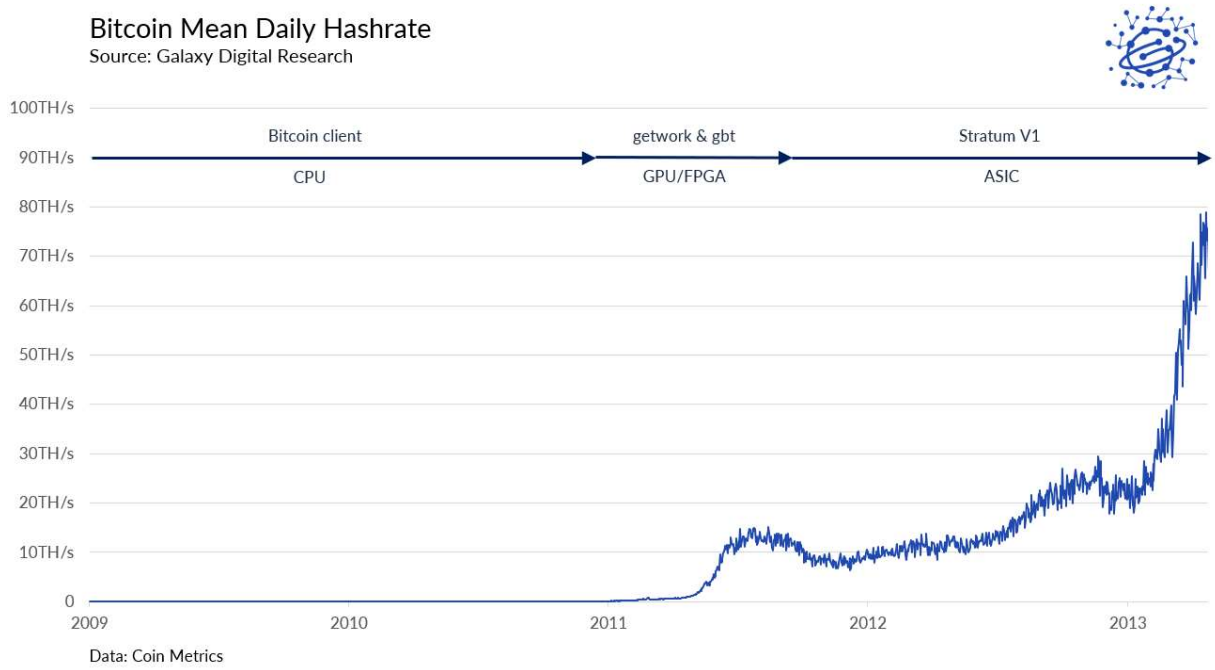

Figure 2: Bitcoin Mean Daily Hashrate, showing the progression of Bitcoin mining protocols through May 2013

In early 2011, CPU mining on the Bitcoin client was phased out in favor of higher performing FPGA and GPU machines, mining via the getwork and gbt protocols. These Stratum predecessors were fine for a pre-ASIC era, but by late 2012, the first Bitcoin ASICs were right around the corner, promising a network hashrate increase like never before.

In response, a Bitcoin Talk Forum user named slush developed the Stratum mining protocol which he released in September 2012. Slush gave the following reasoning for proposing the new Stratum protocol:

> "The reason why I designed this protocol and implemented opensource pool server is that current getwork&lp mining protocol has many flaws and can be hardly used in any large-scale setup. Asic miners are probably coming at the end of the year 2012, so bitcoin community definitely needs some solution which will easily scale to tera-hashes per second per pool user…"[1]

[1] Slush. Stratum mining protocol. https://web.archive.org/web/20120921014023/http://mining.bitcoin.cz/stratum-mining. September 2012.

Slush's reasoning supports the idea that timing is very important when opting to change mining protocols, and that the right time is, again, when the network hashrate makes a major and permanent jump.

Network Hashrate

Still almost a decade after slush announced the Stratum V1 protocol, hashrate continues to rise above expectations. Figure 3 shows how the network hashrate has scaled since the advent of Bitcoin ASICs and the adoption of Stratum V1 in early 2013. Again, hashrate has only increased since the inception of the network.

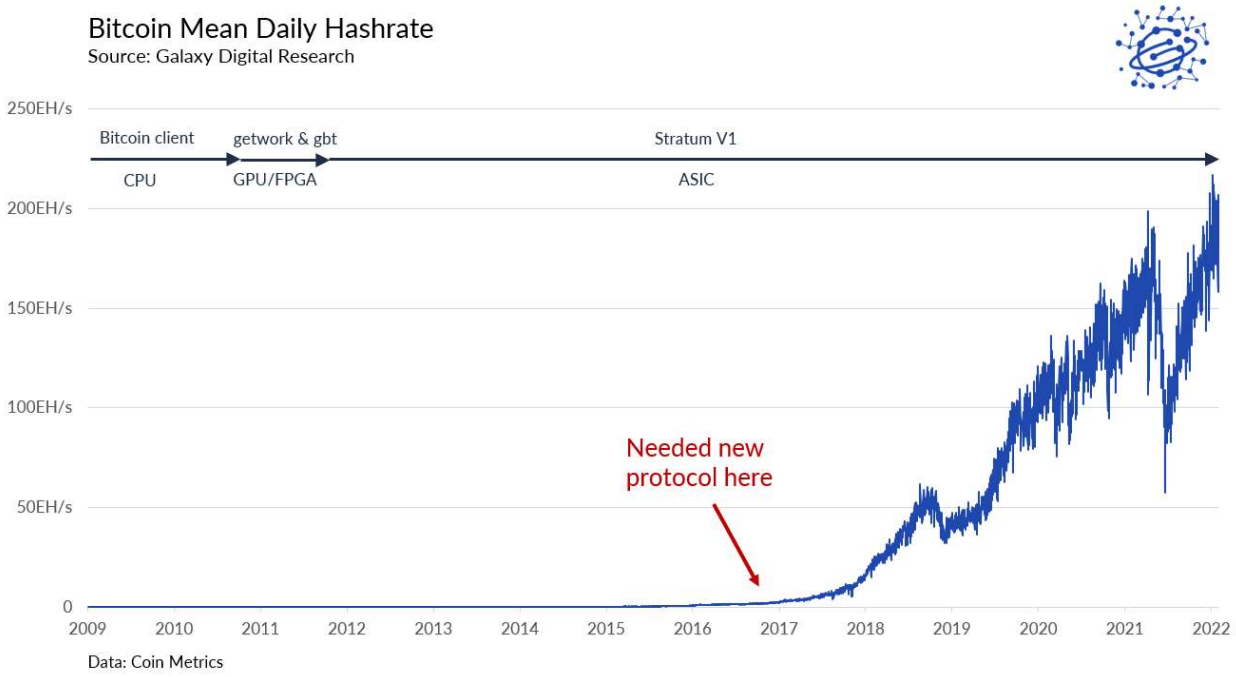

Figure 3: Bitcoin Mean Daily Hashrate (Log), identifying when a new mining protocol should have been implemented

The Stratum V1 protocol is not designed for the high hashrate levels we experience today. When it was originally designed, the network hashrate was only ~12 Th/s. Compare that number to today's hashrate of almost 200 million TH/s. That is an increase of ~16 million times over! Needless to say, things have *changed*, and there are no signs that hashrate growth is slowing down. Galaxy's own hashrate projections forecast a baseline of 335 EH/s by the end of 2022. That is roughly 1.75x the ~189 EH/s of today (January 2022). A *shockingly* quick comeback after China, the previously dominant force in Bitcoin mining, placed a permanent ban on the practice last year.

Based on the network hashrate trend alone, one can easily argue that a new mining protocol is several years past due. In a perfect world, the industry should have received an updated mining protocol in late 2016/early 2017 (indicated on Figure 3) before the sharp runup due to the more efficient machines and greater network adoption. This is incidentally around the time Matt Corallo released his BetterHash BIP draft which was later absorbed into the Stratum V2 protocol[2].

But back to Stratum V1…

Stratum V1 Basics

Stratum V1 Protocol Messages

Traditional solo mining is very much a thing of the past. Today, the vast majority of miners perform pooled mining where the ASIC machines in a mining farm perform the hashing and the pool server sends out the jobs to the machines, with the rewards split

[2] Matt Corallo. BetterHash Mining Protocol(s). https://github.com/TheBlueMatt/bips/blob/betterhash/bip-XXXX.mediawiki. March 2018.

among the pool participants proportional to their compute power. The ASIC machines and pool server communicate by sending messages back and forth. These messages are defined by the Stratum V1 protocol. The four main messages of the protocol are: subscribe, authorize, notify, and submit.

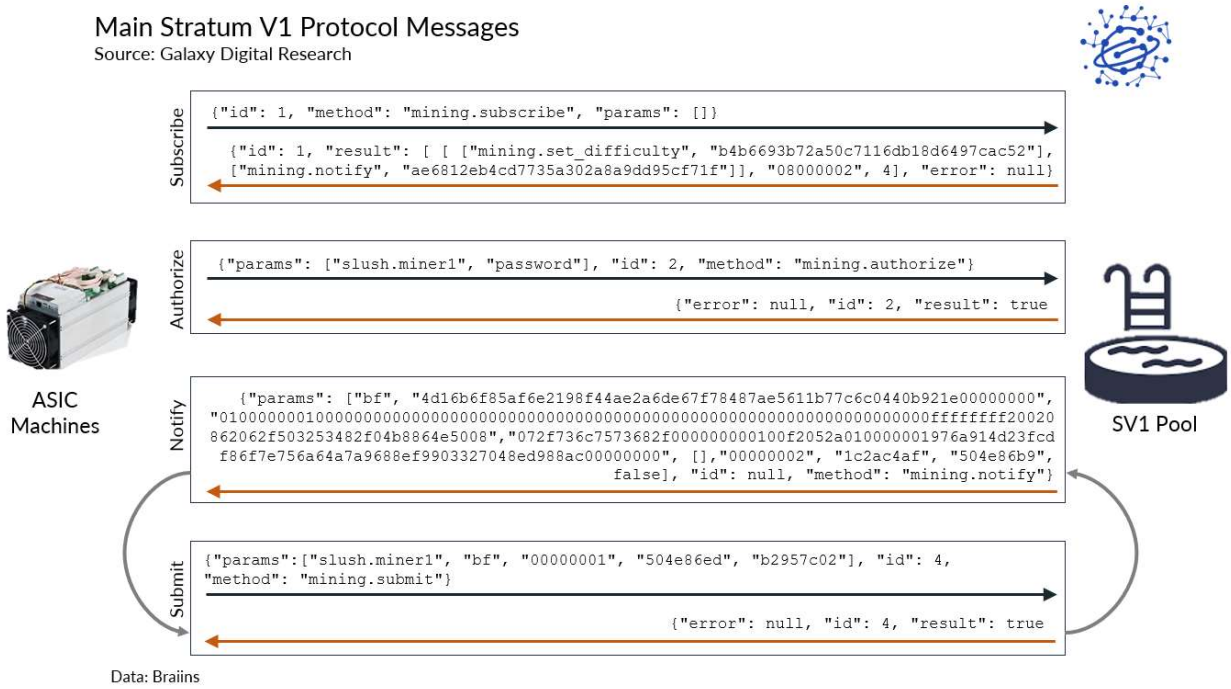

Figure 4: Main Messages of the Stratum V1 Protocol[3]

Figure 4 above depicts the flow of the main messages between the ASIC machines and the pool. The messages shown here are an accurate representation of the actual data that is sent between the miner and the pool in the Stratum V1 protocol.

1. The ASIC machine sends a subscribe message to the pool indicating it wants to authorize itself as a worker and commence mining. If this message was successful, the pool responds with an acknowledgment. Notice that this response contains unnecessary data that cannot yet be used by the miner.
2. The ASIC machine sends an authorize message containing its username and password login credentials for the pool. The pool responds with an acknowledgement. Notice that the password is sent over the wire in plaintext (not encrypted).
3. The pool immediately sends the ASIC machine a notify message that contains all the information required for the miner to commence work.
4. Once the ASIC machine has found a valid job[4], it sends the information to the pool via a submit message. The pool responds with an acknowledgement and then another notify message, continuing the cycle.

Stratum V1 Connection

Figure 5 below illustrates that, as prescribed by Stratum V1, each miner has its own direct connection (or in Stratum V2 term's its own *channel*) to the pool. This is not ideal for the large-scale, industrial mining of today because of the sheer number of connections required in these operations. Mining farms can have tens of thousands of miners and supplying a direct connection between each miner to the pool unnecessarily eats up bandwidth and misses opportunities to share common data between miners. These inefficiencies make a mining operation less efficient, negatively impacting a miner's bottom line.

[3] Braiins. Stratum V1 Docs. https://braiins.com/stratum-v1/docs.
[4] A valid job in this context is one that meets the pool difficulty, which is an easier target to hit compared to the network difficulty. The pool difficulty is used by the pool service to properly estimate a miner's payout.

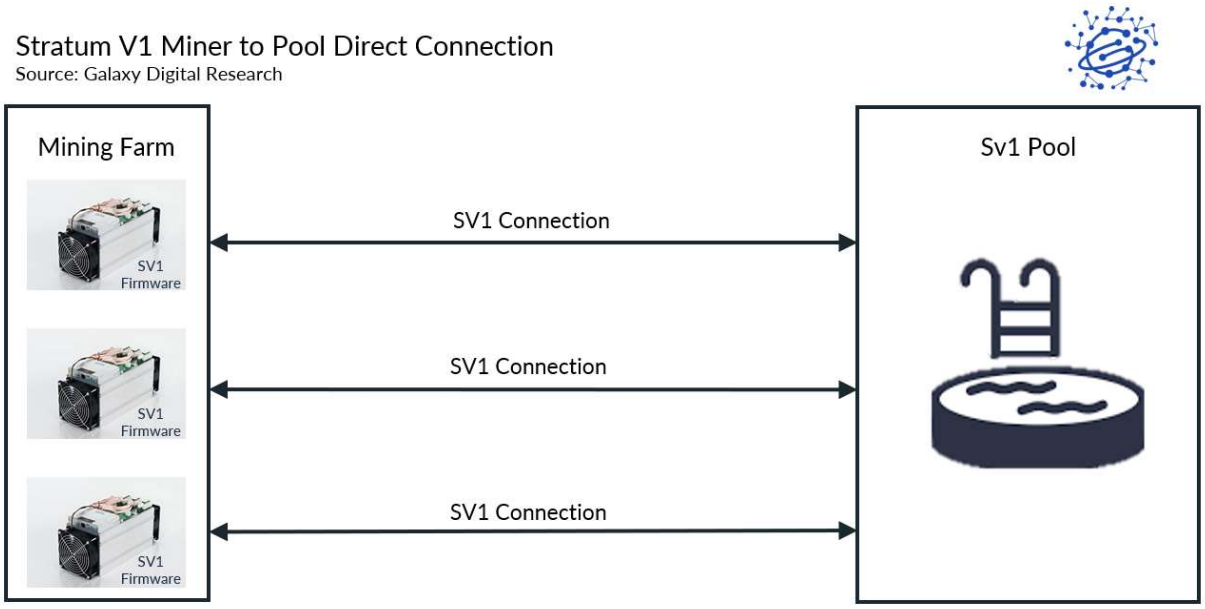

Figure 5: Stratum V1 Miner to Pool Direct Connection

To circumvent this, mining farms use proxy servers to improve the efficiency of their operation. This proxy server is not included in the Stratum V1 protocol, however, and was never standardized. Some proxy servers exist on GitHub but are not very robust as most mining farms develop in-house solutions. How a proxy server typically fits into a mining operation is detailed in Figure 6 below.

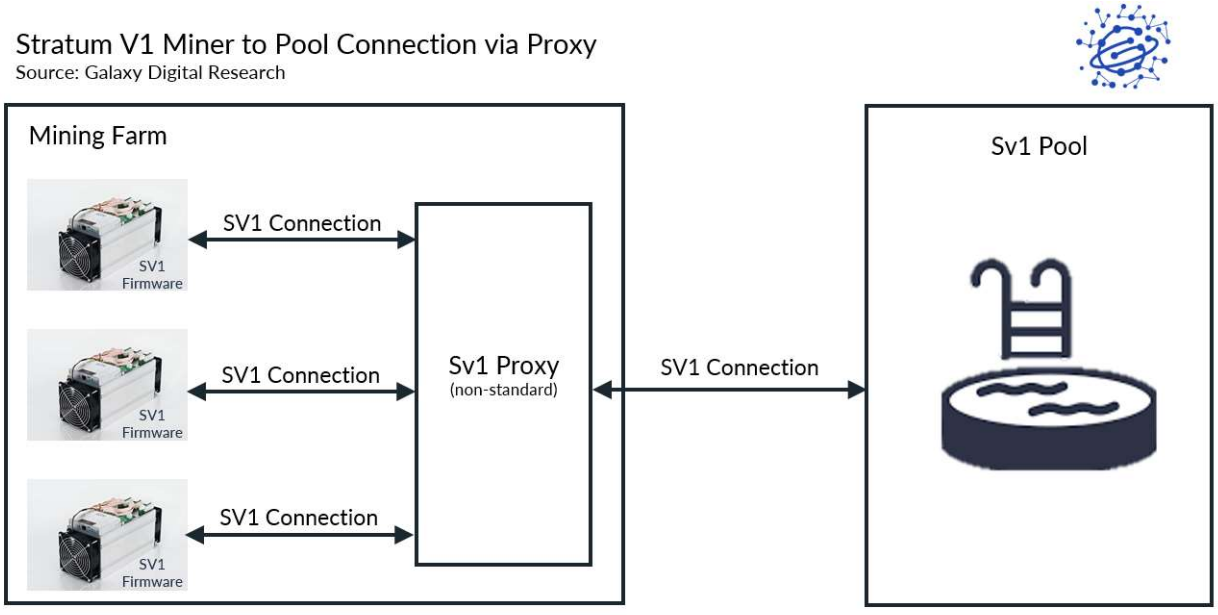

Figure 6: Stratum V1 Miner to Pool Connection via Proxy

Characteristics of a Mining Protocol

Understanding the basics of Stratum V1 helps in determining what characteristics are needed for a mining protocol to better serve today's industry conditions.

The four key features to consider are:

1. Usability
2. Miner Revenue
3. Miner Security
4. Network Security

Metric 1: Usability

First and foremost, a mining protocol must be usable by the miners. It needs to be:

- **Easy to install.** Having a low barrier to entry for miners to commence mining is especially important in regard to the adoption of a new mining protocol in a timely manner.
- **Simple to operate once it is installed.** Complexity in operation or maintenance will eat up miner resources and make adoption less likely.
- **Standardized.** Standardization among setups will make it easier for miners to deploy, maintain, and operate as simply as possible. Standardizing all aspects of the mining protocol ensures that all participants are operating on compatible hardware and are benefitting from the most efficient implementation.
- **Flexible.** The protocol must be suitable for a hobbyist running a single S9, while also scalable to support an industrial grade operation with tens of thousands of machines.

An extra note on flexibility: Flexibility is of utmost importance here. A mining protocol suitable for today must be flexible to fit a variety of miner needs. One way to increase flexibility is by providing miners with different channel configuration options to connect to the pool. Channels are discussed thoroughly below, but for now can be thought of as a connection type that can pass certain types of messages allowing for miners to use different features of the mining protocol. Again, it is beneficial to aggregate these connections to share data where appropriate, cutting down sending unnecessary data to reduce CPU load and bandwidth consumption, ultimately resulting in reduced energy consumption and a higher miner reward.

Metric 2: Miner Revenue

The Miner's main goal is to maximize revenue. Minimizing the computational power and time it takes for a miner to receive a job, commence mining, and send the result back to the pool will benefit miner profitability. Essentially, the protocol needs to make every Watt count.

Designing a protocol that minimizes the number of the messages passed between the miner and the pool, and also minimizes the size of the messages themselves, achieves significant efficiency gains. This saves on the CPU load, bandwidth, and time of both the miner and the pool, reducing a miner's stale job ratio and increasing their reward.

Metric 3: Miner Security

After maximizing a miner's revenue comes keeping that revenue safe. Two large attack surfaces that exist today are main-in-the-middle (MitM) and pool skimming.

Man-in-the-Middle (MitM) Protection

A mining protocol should have the proper encryption and authentication in place such that their reward is protected against a specific MitM attack called hashrate hijacking. During a hashrate hijack attack, the attacker monitors the connection between the pool and the miner and intercepts the packets passed between the two. Unbeknownst to the miner, the attacker replaces the miner's pool login credentials with their own, effectively rerouting the honest miner's hashrate to the attacker.

These attacks can and do happen. In fact, they are probably more common than we think because the attack is very difficult to detect if a mining farm is not explicitly watching for it, a task that is particularly complicated for large mining farms with thousands

of miners. One could even argue that mining farm operators would not publicize this type of attack, as it does not paint them in a favorable light.

Pool Skimming Protection

Pool skimming is another attack vector that is carried out by the pool. In this attack, the pool secretly steals some of the miner's payout. In other words, the pool "skims" off the top of the honest miner's reward. For a mining protocol to truly protect a miner's profits, it should include some mechanism to provide transparency around miner payouts.

Metric 4:

Finally comes network security. The industry needs a mining protocol that aligns miner incentivization with the censorship resistance and decentralization guarantees we all expect from the Bitcoin network.

Stratum V1 vs. Stratum V2

Table 1 compares Stratum V1 and V2 against these high-level metrics that all protocols should have. This table illustrates the value that Stratum V2 will bring to mining and the Bitcoin network.

While Stratum V2 suffers from a low barrier to entry standpoint, it offers us so many more compelling features that should not only ease adoption but result in a more robust and secure mining ecosystem moving forward.

Comparing Stratum V1 and Stratum V2
Source: Galaxy Digital Research

Metric	Stratum V1	Stratum V2
Usability		
Low barrier to entry	✅	❌
Well defined	❌	✅
Well documented	❌	✅
Easy to Use	❌	✅
Flexible	❌	✅
Miner Revenue		
Lean messages	❌	✅
Fewest packets	❌	✅
Miner Security		
Authentication	❌	✅
Encrypted connection	❌	✅
Verifiable Payouts	❌	🟡
Network Security		
Censorship resistant	❌	✅

Table 1: Comparing Stratum V1 and Stratum V2

Metric 1: Usability

Low Barrier to Entry

Despite its limitations (or rather because of its limitations) along with how the industry has grown around this protocol, setting up Stratum V1 is simple. Operationally, a miner needs to plug in their machine, input their pool credentials, and they are mining. The reason why it is so simple is because Stratum V1 is the default protocol implemented on the OEM firmware, so no further

modifications need to be made on the miner's end to begin work. But note, the Stratum V1 protocol is limited. One of its biggest limitations is the lack of a proxy specification. So while commencing work is simple, a miner will incur lower efficiency without developing their own proxy server outside of the protocol specification (not so simple).

This "low barrier to entry" metric is arguably the most important metric when it comes to adopting a new protocol in a timely manner. The complexity of building and operating a mining farm is often overlooked, but it is an enormous undertaking with many moving parts, both figuratively and literally. Therefore, from a miner's perspective, making it as easy as possible for them to set up the firmware is of the utmost importance.

Unfortunately, the two major adoption efforts for Stratum V2—which is discussed later in this report—does not meet this criteria… yet. However, StratumV2 offers additional compelling usability advantages beyond the "barrier to entry" metric.

Well Defined & Well Documented

It is imperative that the mining protocol is well defined and well documented. Standardization will ultimately lead to a healthy development environment and better mining products for the community in the long run.

Stratum V1, while better defined than its predecessors, is still not fully specified. As a result, this ill-defined protocol leaves room for interpretation that has led to inconsistent implementations of logic on both the pool server-side and mining client-side. Generally, it is easy to use Stratum V1 in its most basic form, which is mining straight to a pool using the Stratum V1 OEM firmware. Complications arise, however, if a miner's needs outgrow this simple setup. For example, most miners today are running a lot of ASICs and greatly benefit from using a proxy server to help aggregate the connections and cut down on bandwidth and CPU load. In these more complicated setups, the lack of a robust protocol forces the miners to create custom in house solutions. This diverts valuable time and resources away from the farm and there is no guarantee that the farm has implemented the optimal solution.

Stratum V2, on the other hand, aims to be a fully and precisely defined protocol, leaving no room for interpretation. The Stratum V2 protocol encompasses implementation logic for the ASIC machines, a proxy server, and the pool server. The importance of this cannot be stressed enough. The mining protocol has gone from what is essentially a handful of JSON/RPC messages sent between the pool service and the mining device, to a fully-fledged protocol for the mining device, the proxy, and the pool server logic. With a fully specified protocol, miners and pools alike can be confident that all implementations are not only compatible but are also the most efficient. Something the industry needs drastically.

Easy to Use

Stratum V1 is not that easy to use as it requires a lot of hacks to deal with edge cases (including things as simple as proxy implementations as mentioned throughout this post). This is largely in part due to the lack of standardization as mentioned above.

While both maintaining an extra server to host a Stratum V2 proxy and/or installing the Stratum V2 compatible firmware currently has its barriers, once it is installed it is just as easy to use as the OEM firmware miners currently use for Stratum V1, and certainly easier to use than the Stratum V1 OEM firmware with a homegrown proxy solution.

Flexible

Broadly, Stratum V1 is a very inflexible protocol. As seen in Figure 5 above, each ASIC has a dedicated connection to the pool, which is especially bad for large miners as these separate connections increase the CPU load and eat bandwidth. There are no configuration settings defined by Stratum V1 to aggregate connections (like a proxy server). This is where Stratum V2's fully specified and all-encompassing protocol design really shines above Stratum V1.

As previously mentioned, many miners use a proxy to help aggregate connections to reduce CPU load and bandwidth and improve efficiency. That proxy logic, however, is not natively built into the Stratum V1 protocol and therefore is anything but standardized. There are a few open-source Stratum proxies floating around on GitHub, but they are flimsy and don't handle many not-so-edge cases, so miners are typically forced to develop and maintain their own in-house solutions. Even then, miners have limited control over their search space.

Stratum V2's flexibility largely comes from the communication channels. The protocol specification encompasses the logic for the five roles in Bitcoin mining. To describe which way the data is flowing between these roles, they are classified as being *downstream* or *upstream* relatively to each other. For example, the proxy is considered to be upstream from the mining devices and also downstream from the pool service. These roles are:

1. **Mining Device.** The physical hardware performing the hashing. This role is the most downstream.
2. **Pool Service.** The pool organization that the mining device is contributing hashrate to. This role is the most upstream.
3. **Mining Proxy.** The proxy server that sits in between the mining device and the pool service that performs all the message coordinating and aggregation. This role is considered to be upstream relative to the mining device and downstream relative to the pool service.
4. **Job Negotiator.** Receives the transactions to build custom block templates from the Template Provider (which is just the Bitcoin client) and negotiates the use of this template with the pool. This role is often built into the Mining Proxy.
5. **Template Provider.** This is a Bitcoin client that generates custom block templates to be passed to the Job Negotiator for eventual mining.

The details of the Job Negotiator and Template Provider roles are out of scope for this introductory piece, but it is enough to know that these roles allow for the transaction selection to build a block template with.

The downstream devices open communication channels with the upstream device with established connections. Each channel identifies a mining session associated with an authorized user. All channels are independent of each other, but channels can be grouped to share messages broadcast from the server to achieve higher efficiency (e.g., information about a new prevhash).

Figure 7 details a simple diagram of how the roles are divided on the upstream/downstream scale. Note that while the mining devices and the proxy server live in the mining farm and are both considered downstream to the upstream pool server, the mining devices are *more* downstream relative to the proxy.

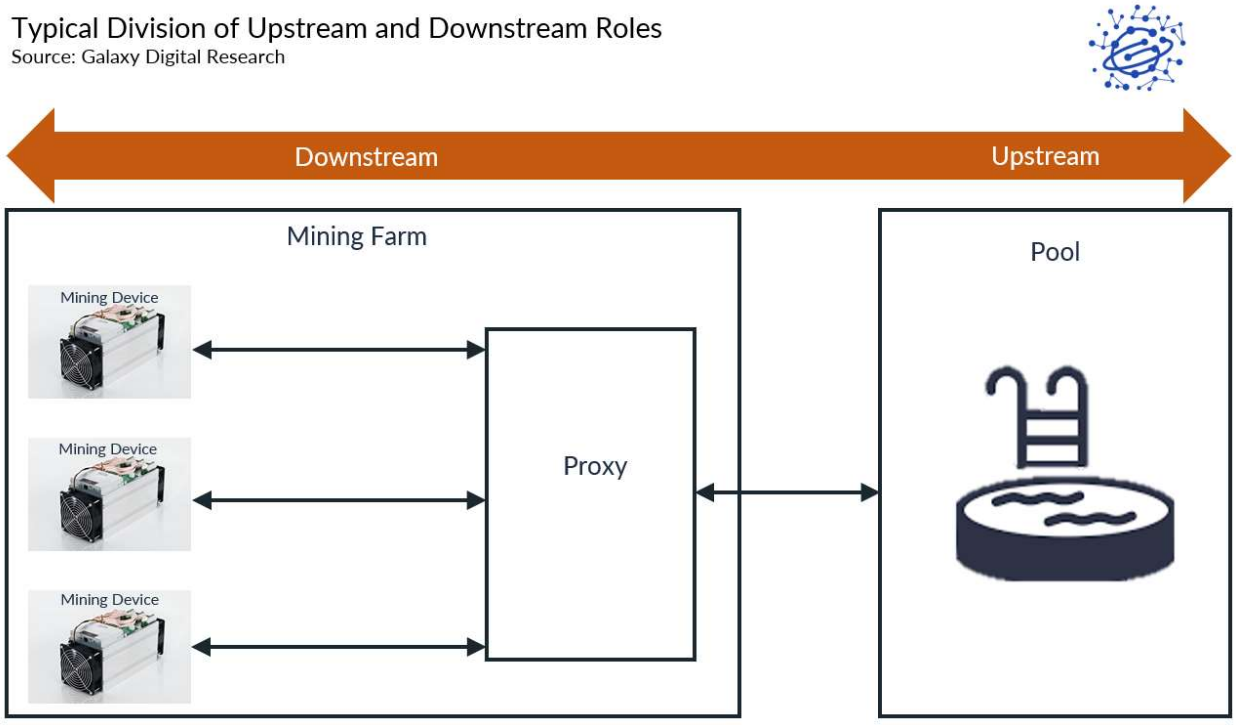

Figure 7: Typical Division of Downstream and Upstream Roles

The proxy can operate in two fashions. In the preferred configuration, the proxy transparently allows its downstream devices (most typically the mining devices but can technically be another proxy server) to open separate channels with the pool server. This requires the mining devices to be running Stratum V2 compatible firmware. In the second, less preferred configuration, the mining devices are running Stratum V1 compatible firmware, and the proxy aggregates these open Stratum V1 connections into its own open channel, translates the Stratum V1 formatted messages to Stratum V2, and forwards them to the pool server.

In both scenarios, the proxy aggregates the connections of the mining devices into a smaller number of TCP connections. Fewer messages are sent in total, leading to lower latency, and larger packets are allowed to be sent, leading to an increase in overall efficiency.

There are three channel types defined by the Stratum V2 protocol that not only allow for the flexible configurations, but also dictate how a miner's search space is prescribed. These channels are:

1. **Standard channels.** These channels are intended to be used between mining devices running Stratum V2 compatible firmware and the proxy server. These channel types can be used between the proxy server and the pool server, but in this case, it is highly encouraged to use group channels (discussed below) to achieve the highest efficiency. A standard channel cannot manipulate the coinbase transaction/Merkle path, as they solely operate on provided Merkle roots. This is called header-only mining (HOM). The guaranteed search space for one standard channel (performing HOM) before nTime rolling is:

$$2^{\text{NONCE\_BITS + VERSION\_ROLLING\_BITS}} = 2^{32 + 16} = \sim 280 \text{ Th}$$

While not as efficient as group channels (defined below), mining with the Stratum V2 protocol using standard channels likely still outperforms mining using the Stratum V1 protocol[5].

2. **Extended channels.** These channels are used between mining devices that are used between the proxy and the pool server. Unlike standard channels, extended channels give the miner extensive control over their search space such that various advanced use cases can be performed. Some of these advanced use cases are difficulty aggregation, custom search splitting, and even translation between the Stratum V1 and Stratum V2 protocols. This translation feature allows for a mining farm to operate Stratum V1 compatible mining firmware on their mining devices, but still gleen some of the efficiency gains realized by the more upstream portions of the Stratum V2 protocol, specifically the connection aggregation performed by the proxy server, and of course the more efficient messaging protocol.

The guaranteed search space for an extended channel before nTime rolling is:

$$2^{\text{NONCE\_BITS + VERSION\_ROLLING\_BITS + EXTRANONCE\_SIZE * 8}}$$

3. **Group channels.** These are used between the proxy server and the pool server. In their namesake, group channels are an aggregation of a set of addressable standard channels into a common communication channel. Group channels are the most efficient form of mining and should always be used if possible.

The guaranteed search space for an extended channel before nTime rolling is:

$$2^{\text{NONCE\_BITS + VERSION\_ROLLING\_BITS + EXTRANONCE\_SIZE * 8}}$$

While each device (mining device, proxy, and pool) must implement at least one channel type to be compatible with the Stratum V2 protocol at some level, not all channel types have to be implemented in order for the protocol to work. Stratum V2 is designed such that only aspects of the protocol can be implemented.

Today, the only Stratum V2 compatible pool is Slushpool. However, only the standard channel type is supported, and to date, no plans to support extended or group channels are in the works. Until pools implement the full feature set of the Stratum V2 pool server logic, it will not be possible to achieve the full efficiency gains that Stratum V2 promises. However, the mere fact that a pool can pick and choose which features they want to implement further highlights the flexibility of this protocol.

Metric 2: Miner Revenue

Maximizing miner revenue is vital for the adoption of a new mining protocol. There must be proper incentivization for the entire industry to switch to a new mining protocol. The best way to make it worth everyone's while is with a promise of increased profits.

Stratum V1 fails on this account largely due to its message design choice which does not minimize the amount of data being transferred between the miner and the pool. This protocol uses verbose, plaintext JSON-RPC messages (as seen in Figure 4),

[5] Hard efficiency numbers are unavailable yet since the protocol is still in development.

which was excellent for initial adoption because it was easy for people to read and understand the protocol. However, mining is *much* too competitive to continue using a human readable message format.

ASIC miners do not natively understand JSON. To execute on these messages, the ASIC wastes time and computational resources serializing and deserializing these JSON messages to and from bytecode. Additionally, Stratum V1 has left over messages from legacy protocols that are irrelevant, wasting computational resources and bandwidth.

Stratum V2 fixes these problems by employing a lean, binary protocol that is machine readable—no more wasting time with JSON. It also removes all unnecessary messages, keeping the amount of data transferred to an absolute minimum, thereby keeping the CPU load and bandwidth consumption to an absolute minimum, and maximizing miner profits.

Metric 3: Miner Security

Miner security is all about keeping a miner's profits safe and whole, something that no pre-Stratum V2 mining protocol has addressed.

Authentication

In general, unauthenticated connections invite MitM attacks, and a mining protocol is certainly not immune. Stratum V1 uses an unauthenticated connection between the miner and the pool, leaving the miner susceptible to hashrate hijacking attacks. Stratum V2 fixes this by simply using an AEAD encrypted channel between the miner and the pool, removing the attack surface entirely.

Verifiable Payouts

Another aspect of miner security is making sure the miners are properly compensated for their work and the pool is not skimming off the top of their rewards. Unfortunately, no mining protocol has addressed this, including Stratum V2. However, Stratum V2 allows for extensions to be added to the protocol. Currently, an effort is underway to *explore* leveraging Schnorr signatures to achieve non-custodial pool services (NC PS), however the extension still does not result in a completely trustless system but rather an improvement[6]. This extension is in its nascent stages of development and further explanation of this extension is out of scope of this blog. However, this is neither completely solve the problem, nor is it officially part of the Stratum V2 protocol. Therefore, in Table 1, the verifiable payout metric for Stratum V2 is marked with a yellow circle to indicate some progress is possible, but metric is not further discussed in this blog.

Metric 4: Network Security

Finally, is network security, something that is unfortunately not optimized for miner's today . There is a major network security vulnerability baked into all pre-Stratum V2 mining protocols: the very real threat of transaction censorship by the pool operators.

Right now, almost all transactions confirmed on the Bitcoin network are done by a handful of pool operators. Individual miners contribute hashes to the pool, but the pool constructs the block template and determines which transactions go inside them. This is **way** too much power in the hands of one group. If pool operators began colluding with each other to censor specific transactions, there is zero recourse the community could take with the current Stratum V1 protocol in place. This is because the Stratum V1 architecture is such that the pool selects the transactions to be included in the candidate block and constructs the block template. Therefore, Stratum V1 is not good for network security because it is incredibly centralized by design.

The full feature set of Stratum V2, however, offers the miner the *choice* to select their own transaction set and construct their own block templates, allowing miners to easily revolt against pools that misbehave on censorship to the detriment of the broader Bitcoin network. Even if miners don't utilize this power, the mere threat that they could, will deter pools from misbehaving.

Adoption Paths

Now that we understand the need for Stratum V2, let's explore the two current adoption efforts:

1. BraiinsOS/BraiinsOS+: A 3rd-party firmware solution from Braiins (the individuals from Slushpool)

[6] https://github.com/stratum-mining/stratum/pull/123

2. stratum-mining: An open source stratum proxy server implementation, written in Rust, that is composed of a set of low level libraries needed to implement Stratum V2 roles, C bindings (to connect to Bitcoin Core, which is written in C), and the implementation of the pool and proxy roles.

The full Stratum V2 protocol encompasses the logic for the ASICs, the proxy server, and the pool server. The Braiins effort is focused wholly on implementing the ASIC protocol logic in the ASIC's firmware, but it does not include the proxy or pool server logic. The stratum-mining effort is focused on implementing the proxy protocol logic which a mining farm operator will install on a separate server (most likely local to their farm). To take full advantage of the Stratum V2 protocol, a mining farm would need to have Stratum V2 compatible firmware (like BraiinsOS) on their ASIC machines and a proxy server running the stratum-mining, while also mining on a pool that supports the group channel connection type (defined in the proceeding section).

In the following subsection, both adoption efforts are discussed in more detail.

Effort 1: BraiinsOS/BraiinsOS+ 3rd-Party Firmware

The individuals at Braiins are doing an outstanding job developing 3rd-party mining firmware that is Stratum V2 compatible and production ready. At the time of writing, it is the only production ready implementation of Stratum V2.

There are two forks of this project: BraiinsOS and BraiinsOS+. BraiinsOS is a completely open source effort and is currently compatible with S9, S9i, and S9j ASIC models. BraiinsOS+ is a fork of BraiinsOS with some closed source auto-tuning features included. BraiinsOS+ is currently compatible with S17, S17 Pro, S17+, S17e, T17, T17+, T17e, S9, S9i, and S9j ASIC models, with M20S and S19 compatible firmware in the development pipeline. BraiinsOS+ is currently running on over a hundred thousand machines, a very impressive feat that should be celebrated by the community.

High Level Configurations: 3rd-Party Firmware

Figure 8 displays the high-level configuration of machines running 3rd-party, Stratum V2 compatible firmware like BraiinsOS. This figure shows that, just like with Stratum V1, there is an independent connection (using standard channels) for each ASIC to the pool.

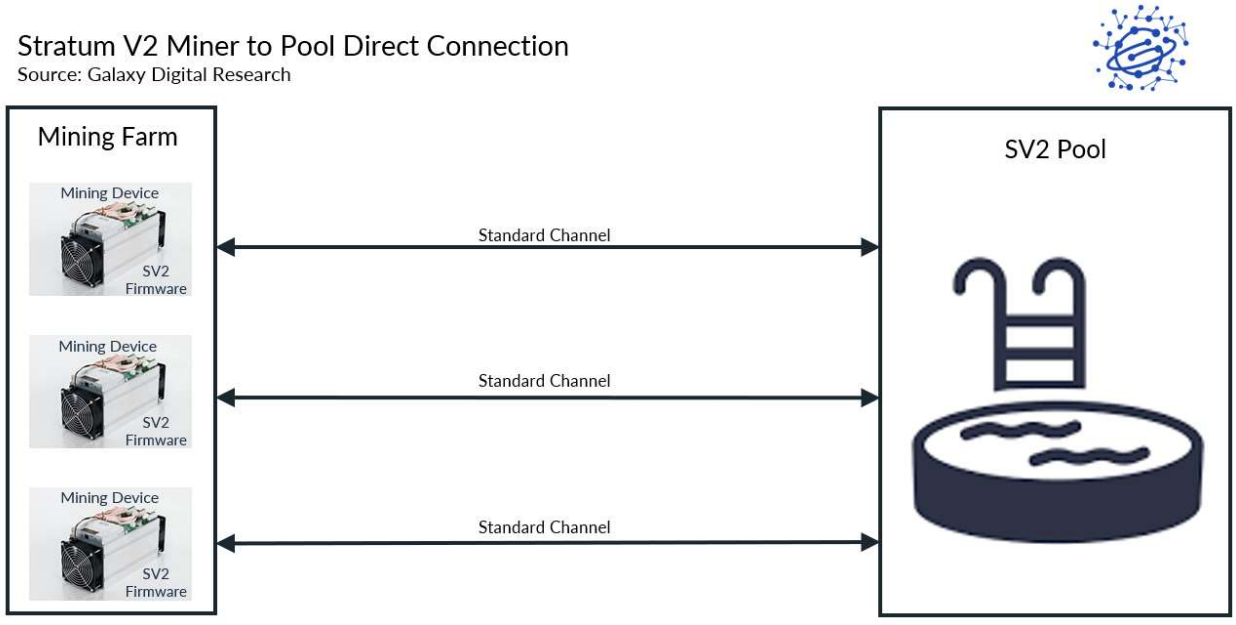

Figure 8: Stratum V2 Miner to Pool Direct Connection

The reader may be asking themselves “if this is operationally the same as Stratum V1, then what is the hold up with adoption?” What hurdles *are* preventing all miners from installing Stratum V2 compatible firmware on their machines?

Hurdles: 3rd-Party Firmware

Firmware development is typically a slow process, especially firmware development for ASICs as the chip architecture varies from model to model. 3rd-party ASIC firmware developers, like the team at Braiins, must constantly assess which ASIC models to prioritize development for. Therefore, when a new machine is released, 3rd-party firmware for a new model typically lags.

Furthermore, ASIC manufacturers want their clients to use their own OEM firmware and have made it increasingly difficult to install 3rd-party *any* firmware—not just BraiinsOS. Machines come preloaded with firmware from the manufacturer, called the OEM firmware. Adding 3rd-party firmware (firmware not made by the ASIC manufacturer) voids the warranty (which is usually one year). All the power here is in the hands of the manufacturer. This is a problem for the industry as a whole and is contrary to the open-source ethos of Bitcoin.

Effort 2: Open-Source Stratum Proxy

The other adoption effort is to build an open source Stratum proxy server, which is still in its development phase. This effort is mainly focused on the development of the Stratum V2 proxy server logic. This proxy server is a very powerful tool for miners as it does all the heavy lifting orchestrating the communication between the ASIC machines and the pool server in an efficient manner. Several companies, including Square Crypto, BitMex, and Galaxy Digital, see the value of Stratum V2 and have dedicated resources to fund development.

High Level Configurations: Stratum Proxy

Figure 9 displays the high-level configuration of a mining operation running Stratum V2 compatible firmware using the Stratum proxy, opting for the Pool Service to perform the transaction set selection. The proxy does all the heavy lifting here. Notice the different channel types. Again, these channels are what lends inherent flexibility to the Stratum V2 protocol. Machines are connected to the proxy server via a standard channel. Then a proxy is connected to the pool via a group channel, which aggregates the standard channels into one connection to the pool. This architecture results in data being passed more efficiently, reducing the CPU load and bandwidth consumption, increasing miner revenue.

Note that, once the development of this Stratum proxy is complete, the architecture displayed in Figure 9 will be the most efficient form of mining.

Stratum V2 Miner to Proxy Connection Via Standard Channels, Proxy to Pool Connection via Group Channel, With Pool Selected Transaction Sets
Source: Galaxy Digital Research

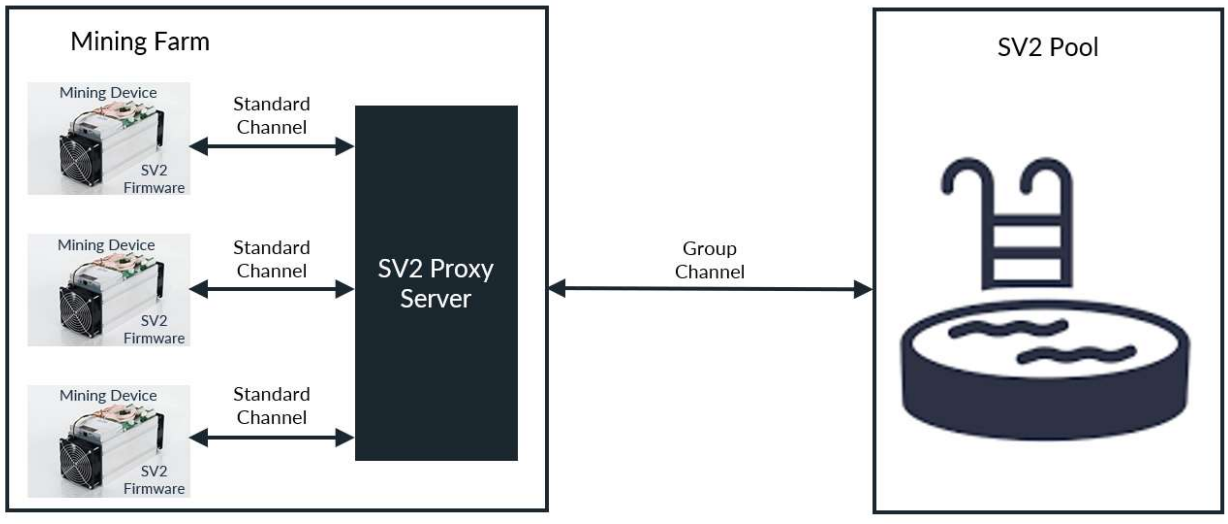

Figure 9: Stratum V2 Miner to Proxy Connection via Standard Channels, Proxy to Pool Connection via Group Channel, with Pool Selected Transaction Sets

A big difference in this implementation is the miner selected transaction set. Figure 10 below displays the high-level architecture of a mining operation running Stratum V2 compatible firmware using the Stratum proxy, opting for themselves (the Miner) to perform

the transaction set selection. From an architecture perspective, all the miner needs to do is deploy a bitcoind node and the stratum proxy handles the rest. So simple!

Note: From a developer's perspective there is a lot more going on under the hood when it comes to miner selected transaction sets, but the important thing is that it is simple from a miner's point of view.

Stratum V2 Miner to Proxy Connection Via Standard Channels, Proxy to Pool Connection via Group Channel, With Miner Selected Transaction Sets
Source: Galaxy Digital Research

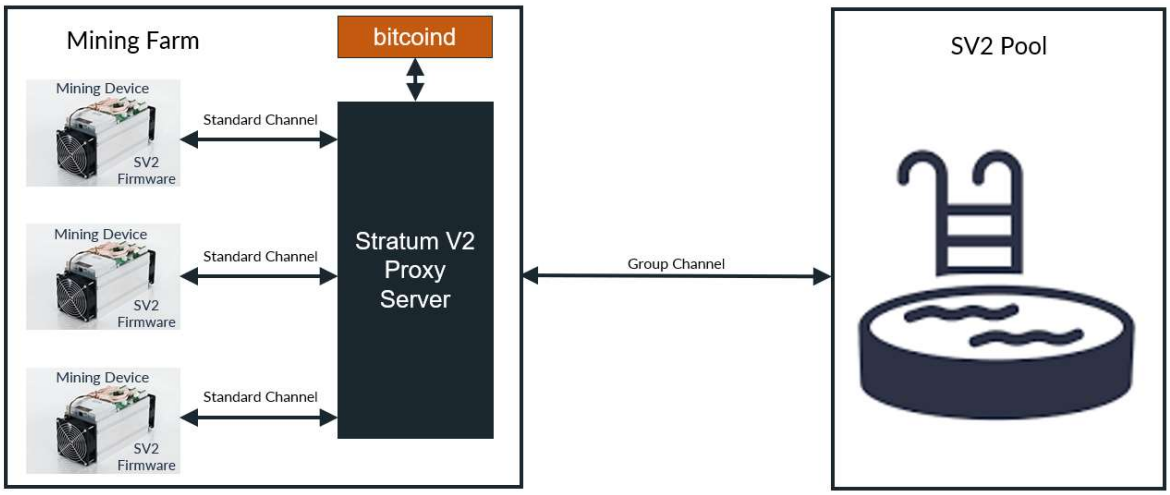

Figure 10: Configuration of the Stratum V2 Compatible Machines Connected to the Stratum Proxy with Miner Selected Transaction Sets

So far, the configurations explored in this section all require the ASIC machines to have Stratum V2 compatible firmware installed, which as previously explained, is currently easy to do. One very useful feature of the Stratum proxy is that it works with machines running Stratum V1 firmware. This eases adoption as a vast majority of machines today are running Stratum V1 firmware, giving these Miner's the ability to use Stratum V2 before making the firmware switch further down the line.

Figure 11 shows this configuration. While it is not as efficient as the configuration shown in Figure 9 and 10, it is much more efficient than using a Stratum V1 proxy or no proxy at all.

Stratum V1 Miner to Proxy, Proxy to Pool Connection via Extended Channel, With Pool Selected Transaction Sets
Source: Galaxy Digital Research

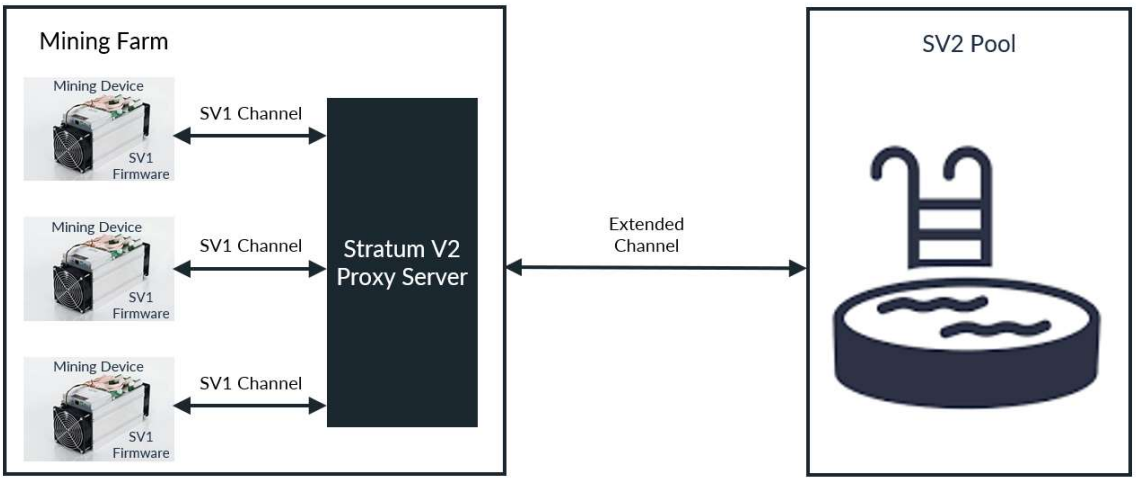

Figure 11: Stratum V1 Miner to Proxy, Proxy to Pool Connection via Extended Channel, with Pool Selected Transaction Sets

Figure 11 Stratum V1 Miner to Proxy, Proxy to Pool Connection via Extended Channel, With Pool Selected Transaction Sets

And even with Stratum V1 compatible firmware, a Miner can still choose their own transaction sets! Talk about flexibility!

Figure 12 shows the same configuration displayed in Figure 11, only includes a bitciond node.

Stratum V1 Miner to Proxy, Proxy to Pool Connection via Extended Channel, With Miner Selected Transaction Sets
Source: Galaxy Digital Research

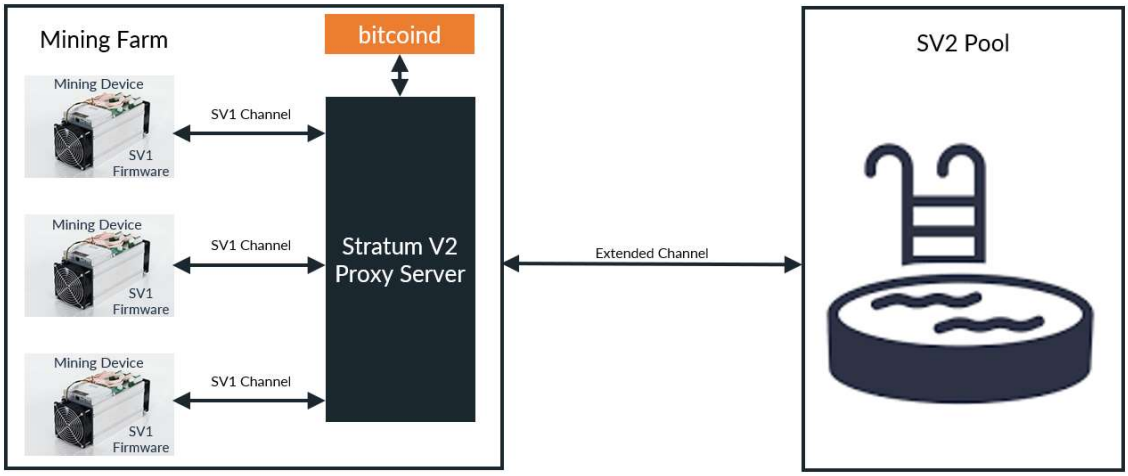

Figure 12: Stratum V1 Miner to Proxy, Proxy to Pool Connection via Extended Channel, with Miner Selected Transaction Sets

Hurdles: Stratum Proxy

By design, the Stratum proxy requires the operation of an additional server. For a miner to build their own block templates, a Bitcoin node must also be maintained by the miner (as seen in Figures 10 and 12 above). While this is not a big ask, it is still an ask. And any task in this industry is an extra burden of entry for the average miner that can impede adoption.

Adoption Efforts Assessment

Now that we are aware of the two major Stratum V2 adoption efforts (firmware and proxy), let's use the previously established metrics to compare these efforts with the Stratum V1 protocol. Table 2 displays the full comparison.

Comparing Stratum V1 and Stratum V2 Implementation Styles
Source: Galaxy Digital Research

Metric	Stratum V1	BRAIINS OS	stratum-mining / stratum
Usability			
Low barrier to entry	✅	❌	❌
Well defined	❌	✅	✅
Well documented	❌	✅	✅
Easy to Use	❌	✅	✅
Flexible	❌	❌	✅
Miner Revenue			
Lean messages	❌	✅	✅
Fewest packets	❌	❌	✅
Miner Security			
Authentication	❌	✅	✅
Encrypted connection	❌	✅	✅
Verifiable Payouts	❌	❌	🟡
Network Security			
Censorship resistant	❌	❌	✅

Table 2: Comparing Stratum V1 and Stratum V2 Implementation Styles

The metrics that the adoption efforts fall short of meeting will be the focus here.

Both the 3rd-party firmware and the proxy have a higher barrier to entry compared to the Stratum V1 firmware: 3rd-party firmware is difficult to install because of how the ASIC manufacturers operate, so it falls short on the low barrier to entry metric, and the Stratum proxy requires the installation and maintenance of an addition server by miners, which makes setup more difficult.

The *full* feature set of Stratum V2 (something that is still in its development stage) successfully fulfills the remaining metrics, however 3rd-party firmware solutions still suffer from inflexibility, efficiency, and censorship resistance. 3rd-party firmware without a proxy is rigid in its Miner-Pool Service connection, which negatively impacts efficiency. However, it is still *much* more efficient than using Stratum V1. In fact, while efficiency is hard to measure because of all the moving parts, Braiins estimates a ~35% increase in overall efficiency in comparison to the OEM's Stratum V1 firmware.

Furthermore, 3rd-party firmware also does not include the logic to allow miners to choose their own transaction set without the use of the Stratum proxy, leaving the miner (and network) exposed to transaction censorship by pools.

Overall, both adoption efforts offer many more benefits than the legacy Stratum V1 protocol, but perhaps another adoption effort is possible, one that will not give us everything we want, but will be better for the initial, widespread adoption of Stratum V2.

Stratum V2 OEM

The two Stratum V2 development efforts, while superior to Stratum V1 in nearly every other way, require extra effort on the miner's part to set up and, in the case of the Stratum proxy, require an additional server. What would be the nail in Stratum V1's coffin is if miners could seamlessly switch to the Stratum V2 protocol. Convincing ASIC manufacturers to include Stratum V2 in their OEM firmware could make that possible. If getting setup and running with Stratum V2 is functionally identical to Stratum V1, Stratum V2 will be unquestioningly and quickly adopted.

Table 3 shows a comparison of just firmware, specifically, the Stratum V1 OEM firmware, the theoretical Stratum V2 OEM firmware, and Stratum V2 3rd-party firmware. Notice that the theoretical Stratum V2 OEM firmware fulfills that low barrier to entry metric, which is excellent for adoption. With this, the transition from Stratum V1 to Stratum V2 firmware is functionally the same from the

miner's perspective: they receive the ASIC in the mail pre-configured with Stratum V2 firmware or they do a routine firmware update. BOOM! Stratum V2 is the new industry standard.

Except for ease of adoption, this theoretical Stratum V2 OEM firmware still suffers from the same shortcomings as 3rd-party firmware when it comes to flexibility and efficiency in comparison to using a Stratum proxy (although it is still more efficient than the Stratum V1 OEM firmware). It also does not give the miners the option to select their own transaction set. But there is a strong argument to be made that sacrificing these features in the short term in favor of Stratum V2 implementation results in overnight adoption.

We need to walk before we can run with this protocol as it brings about many changes that will require miners and pool services to adjust. The first step in achieving the ultimate goal of gaining transaction censorship resistance is getting all the Miners to use Stratum V2 in its most basic form. As much as the Bitcoin community wants every miner to select their own transaction sets, this feature needs to be looked at like a safety switch for the network: Miners are able to use it as a fallback if there is suspicion of nefarious pool activity. If pools start censoring transactions, every miner already has the infrastructure in place to switch to their own transaction set, avoiding the scenario where Miners have no recourse if/when this becomes an active threat.

Comparing Stratum V1 and Stratum V2 Firmware Types
Source: Galaxy Digital Research

Metric	SV1 OEM Firmware	SV2 OEM Firmware	BRAIINS OS
Usability			
Low barrier to entry	✅	✅	❌
Well defined	❌	✅	✅
Well documented	❌	✅	✅
Easy to Use	❌	✅	✅
Flexible	❌	❌	❌
Miner Revenue			
Lean messages	❌	✅	✅
Fewest packets	❌	❌	❌
Miner Security			
Authentication	❌	✅	✅
Encrypted connection	❌	✅	✅
Verifiable Payouts	❌	❌	❌
Network Security			
Censorship resistant	❌	❌	❌

Table 3: Stratum V1 OEM Firmware vs. Theoretical Stratum V2 vs. 3rd-Party Stratum V2 Metrics Comparison

It may sound simple to get the ASIC manufacturers to include Stratum V2 in their OEM firmware but convincing an ASIC manufacturer to modify their firmware in any way is *much* easier said than done. It is common knowledge in the mining space that ASIC manufacturers are *fiercely* protective over their firmware, and certainly do not want miners using 3rd-party firmware solutions.

But there could be a way to speed up the adoption of Stratum V2 by the manufacturers that does not require a complete rewrite of their firmware. It is widely rumored that the ASIC manufacturer's OEM firmware is a closed source fork of the open source cgminer, which is under the GPL-3.0 license. Something that–putting it mildly–is not so great for the industry. However, this piece of information could be used to Stratum V2's advantage. An open source effort could be started where cgminer is forked yet again and the Stratum V2 logic is then implemented in the fork, which is left as open source in accordance with the license. Because the manufacturer's firmware code is already a fork of the cgminer code, it is possible to merge this theoretical Stratum V2 cgminer fork into the manufacturer's closed source fork since they have some shared history (up until a certain point). This is of course contingent upon the manufacturer's desire to support Stratum V2.

The rigmarole of getting ASIC manufacturers to include Stratum V2 into their firmware highlights an interesting underlying dynamic at play: the mining industry is very divided between what the manufacturers want and what the developers want. This disconnect is especially noticeable when it comes to feature prioritization of a mining protocol.

Ultimately, Bitcoin is governed by miners, developers, and nodes, and each may have different priorities but together form a *governance triumvirate* with each constituency providing checks on the power of others. But miners should never have to prioritize decentralization and censorship resistance above their bottom line. A miner's job is *not* to ensure the censorship resistance of the network. A miner's job is to be *incentivized* to *secure* the network. It is up to the developers to provide a mining protocol that aligns censorship resistance with miner incentivization. Stratum V2 is that effort, and it will improve Bitcoin for all who use it.