Theta Mainnet 3.0 Whitepaper

Theta Labs, Inc

December 2020

Table of Contents

1. Introduction

Theta Labs launched its native blockchain purpose-built for peer-to-peer video delivery in the Spring of 2019. With Mainnet 2.0 in May 2020, we introduced Guardian nodes, a revolutionary, two layer consensus mechanism [1, 2] to complement its Enterprise validators run by a premiere set of global partners including Google, Samsung, Binance, Blockchain.com, and Gumi.

Following Mainnet 2.0, the capability of the Theta Edge Node network was significantly enhanced with the release of Theta's latest peer-to-peer "EdgeCast" technology [3]. This new fully decentralized technology stack adds the ability to capture live video, transcode it in real-time, cache and relay live stream video data to users globally - all through Theta's P2P

edge network run by thousands of community members. Not a central server or service is used in this pipeline.

Additionally, the long-awaited Turing-complete smart contract support was enabled on the Theta Blockchain Mainnet in Q4 2020. Smart contracts open up a whole new set of user experiences and new attribution models for DApps built on the Theta network. For example, leveraging smart contracts on the Theta network could enable fully digitized item ownership, innovative payment-consumption models, transparent royalty distributions, trustless crowdfunding mechanisms, and much more. This provides an additional layer of social and economic interactivity that supplements the core functionality of video and data delivery, and significantly increases the engagement and retention of platform users.

Building on this foundation, we are thrilled to announce Theta Mainnet 3.0, **estimated to launch in the Spring of 2021**, and introduce two novel protocol innovation:

1. Upgraded **Elite Edge Nodes** will enable "Uptime Mining", the tokenization of Internet bandwidth and availability. Users will be able to stake TFuel to an edge node to upgrade it to an Elite Edge Node. Elite nodes can 1) earn TFuel through their staked TFuel, and 2) earn additional TFuel from video platforms for delivering higher performance through a "Proof-of-Relay" mechanism.

   In the process of engaging with platform partners, it became clear that to support an existing platform that may see over 1 Billion logged in users per month or even a smaller partner with tens of millions of subscribers, requires an edge delivery network that needs to scale to tens and hundreds of thousands of nodes globally to deliver premium service levels, uptime and performance across all devices. We believe Uptime Mining and Proof-of-Relay are effective incentivization mechanisms for boosting the footprint of the edge network to meet the needs of major video platforms.

2. New **TFuel burn** will add a cost for using Theta edge network, namely a "network fee". Since Theta network launched two years ago we recognized that there is significant value accrued to video platform partners in the form of content delivery network (CDN) cost savings, increased user engagement, and revenues resulting from shifting infrastructure costs to user rewards. All these benefits accrue to the platform partner at effectively zero additional cost. With Theta 3.0, a portion of each TFuel **payment to the edge network** will be burned at the protocol level, effectively becoming a cost of using the edge network. In the long-run as Theta's edge network becomes more widely adopted, this could meaningfully reduce the supply of TFuel. Note that the TFuel burning concept is similar to the Ethereum EIP-1559 proposal which introduces "BASEFEE burn" as a deflationary mechanism to the ETH supply. That mechanism adds to the scarcity of ETH and ensures the long-term security of Ethereum [7].

2. Enhanced Network Economics

The overarching goal of the Theta crypto economics design is to properly incentivize and reward all Theta ecosystem stakeholders, and thus ensure the security and utility value of the Theta network.

Theta Mainnet 2.0 introduced a TFuel inflation reward of 5% per year for Theta staking as shown in Diagram 1 below, which incentivizes token holders to stake Theta to validators and guardian nodes, and thus secures the blockchain network through Theta's multilevel-BFT consensus mechanism. In addition, video platforms purchase TFuel in order to incorporate Theta P2P video delivery protocol into their infrastructure, reward their end-users for sharing bandwidth and as a payment method to content creators.

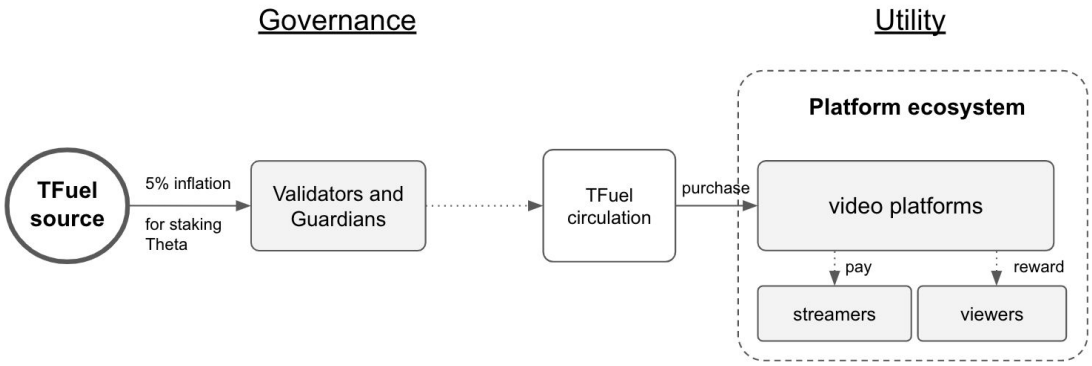

Diagram 1. Theta crypto economics as of today.

In Mainnet 3.0, we extend the protocol to reward Theta edge nodes, another important network participant responsible for providing video delivery services as shown in Diagram 2. Enhanced economics include new TFuel inflation, staking and burning.

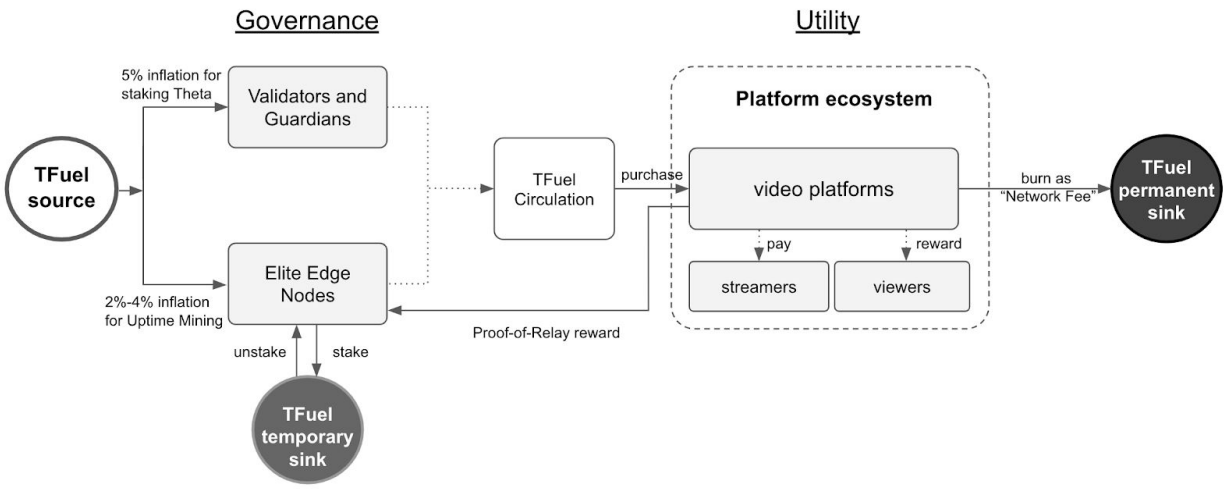

Diagram 2. Theta Mainnet 3.0 crypto economics.

1. **New 2%-4% TFuel inflation** for uptime mining for edge nodes will be introduced in addition to existing TFuel rewards for validators and guardians. All edge nodes will have the option to stake **TFuel** to upgrade itself to an **Elite Edge Node**, and earn its portion of the newly inflated TFuel through the "Uptime Mining" mechanism to be described in Section 3. The additional inflation will be split proportionally among elite nodes calculated based on the amount of TFuel staked, and the uptime score of each node. We believe that with proper inflation incentives, the number of edge nodes can quickly grow from thousands to tens or even hundreds of thousands, which can provide sufficient bandwidth to cater to the needs of major video platforms with millions of users. While this approach is effective in expanding the capacity of the edge network, inflation alone does not retain or increase the utility value of TFuel. This is especially true when the amount of TFuel generated exceeds the demands from the video platforms and end viewers.

2. **TFuel burning as a balancing force** will be introduced. The Elite Edge Network will enforce a rule that **at least 25%** of each TFuel payment to the network will be burned, treated as network usage fee. To see why burning can serve as an effective mechanism to sustain the utility value of TFuel, we can refer to Irving Fisher's equation of exchange $MV = PT$, where M represents the circulation supply of a token, and V represents the velocity of the token (i.e. the average frequency with which one unit of token is spent) [4]. This equation states that higher token velocity requires lower token circulation supply when other conditions remain the same. Conceivably, with a portion of TFuel burned for each edge network payment, the supply of TFuel can be reduced gradually, and reduced at a faster pace when the network usage increases. Eventually any excess amount of TFuel will be burned, and if at that point the TFuel inflation rate (a blockchain parameter) can be adjusted to match the burning rate, the system will enter a balanced state where

the TFuel utility value is effectively stabilized, an example for illustration purposes shown in Diagram 3 below.

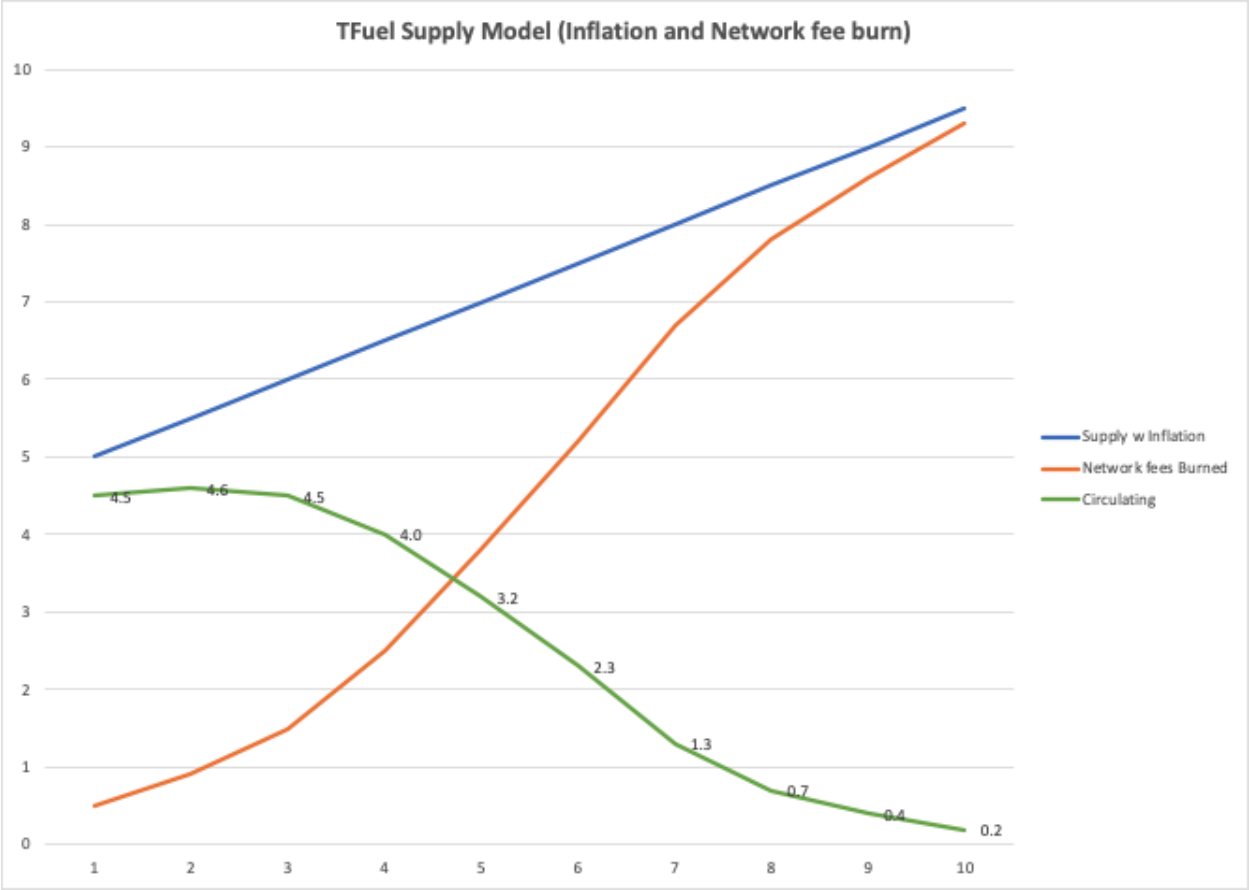

Diagram 3. Example of TFuel inflation, burning, and circulation (for illustration purposes)

In short, Theta 3.0 enhances its crypto economics to provide both **sources and sinks for TFuel** at the protocol level, as shown in Diagram 4 below:

1. TFuel inflation acts as the source. Validators and Guardian Nodes earn a part of the inflated TFuel for securing the blockchain network. Elite Edge Nodes earn the remaining portion through Uptime Mining for being available to provide useful services.

2. TFuel burning becomes a permanent sink which takes the burned TFuel perpetually out of circulation. In addition, in the Theta blockchain, transaction fees and smart contract gas fees simply vanish instead of being paid to the validators. Therefore, they are two other permanent sinks for TFuel.

3. TFuel staking to elite nodes can be viewed as a temporary sink which takes TFuel out of circulation for a period of time.

4. At the application level, with Turing-complete smart contract support, potentially more temporary sinks for TFuel can be introduced, such as liquidity pools for Uniswap-like AMM DEX, asset collaterals for MakerDAO-like stable currency, etc.

With both sources and sinks in place, TFuel circulation can be adjusted dynamically to adapt to actual demand. We believe this can maximize the utility value of TFuel and benefit all stakeholders of the Theta ecosystem.

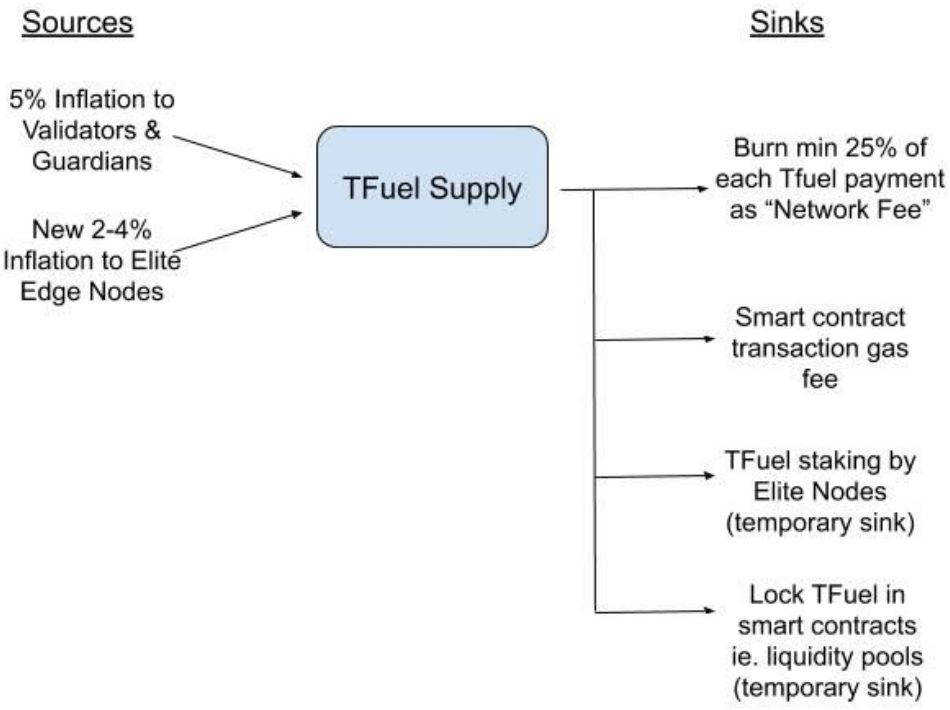

Diagram 4. Sources and sinks for TFuel.

3. Elite Edge Node and Uptime Mining

Dual Rewards to Elite Edge Nodes

With Mainnet 3.0 any user can install the Edge Node software as before, but with an option to **stake TFuel** to upgrade itself to an **Elite Edge Node**. Elite nodes earn a portion of the newly inflated TFuel based on the amount of TFuel staked and its uptime score. It has no dependency on the amount of data relayed by the node.

In an employment analogy, this can be viewed as the **"base salary"** for an elite node. On top of this base salary, if the node relays data it is also eligible to earn more TFuel through our **"Proof-of-Relay"** mechanism detailed in Section 5 which extends Mainnet 2.0 design. However, this TFuel does not come from the protocol-level TFuel inflation, but is paid by platform partners directly.

Additionally, for each platform payment to the network, a portion will be burned and the remainder would be split among the Elite nodes that submit a Proof-of-Relay. This portion of the TFuel reward can be viewed as the **"performance bonus"** based on the actual bandwidth and services contributed to platform partners. As before, video platforms can optionally reward their end-viewers and streamers with TFuel as any other DApps running on Theta. This way, all stakeholders of the video streaming ecosystem including end viewers, streamers, the platform itself and Theta elite nodes clearly benefit by participating in the ecosystem.

There will be a lower and an upper limit on the amount of TFuel that can be staked to an Elite node. The lower limit is necessary to prevent sybil attacks, will be explained later. The upper limit is to ensure the most optimal level of decentralization. If users want to stake more TFuel than the upper limit, they can launch multiple edge nodes and split their TFuel across those nodes.

Uptime Mining

At the protocol level, the Aggregated Signature Gossip routine [1, 2] used in the multi-level BFT protocol will be enhanced to prove the uptime of the Elite Edge Nodes. Elite Edge Nodes will be connected to Validator and Guardian nodes to form a joint mesh network, as depicted in Diagram 5. On this mesh network, elite nodes broadcast their BLS signature of the latest checkpoint block hash for guardian nodes to aggregate. The BLS signature share of an elite node proves that the node was up and running when the corresponding block was produced. However, to support potentially hundreds of thousand or even millions of Edge Nodes, for each checkpoint block, a subset of elite nodes is chosen by random to receive the inflated TFuel and in the long run, each node will expect to receive its proportional share similar to Bitcoin mining. If the signature share of an elite node is included in a checkpoint, the node will be qualified to split the TFuel reward for that checkpoint proportionally to the amount of the TFuel staked.

To avoid missing TFuel rewards, an elite node would need to stay online all the time, download the latest block header, check if it is in the sampled set, sign the checkpoint block hash and gossip out the signature immediately if selected. This provides incentive for elite node operators to maximize the uptime of their nodes, which is beneficial to the Theta network users since this improves the availability of the network. We thus call this mechanism "**Uptime Mining**", and simply keeping an elite node running and available for data relay and other tasks, the node operator would be eligible to earn the TFuel inflation rewards. We believe with this uptime reward mechanism, the Edge Node network can quickly expand its node count and bandwidth capacity to serve the ever increasing need from video platform partners.

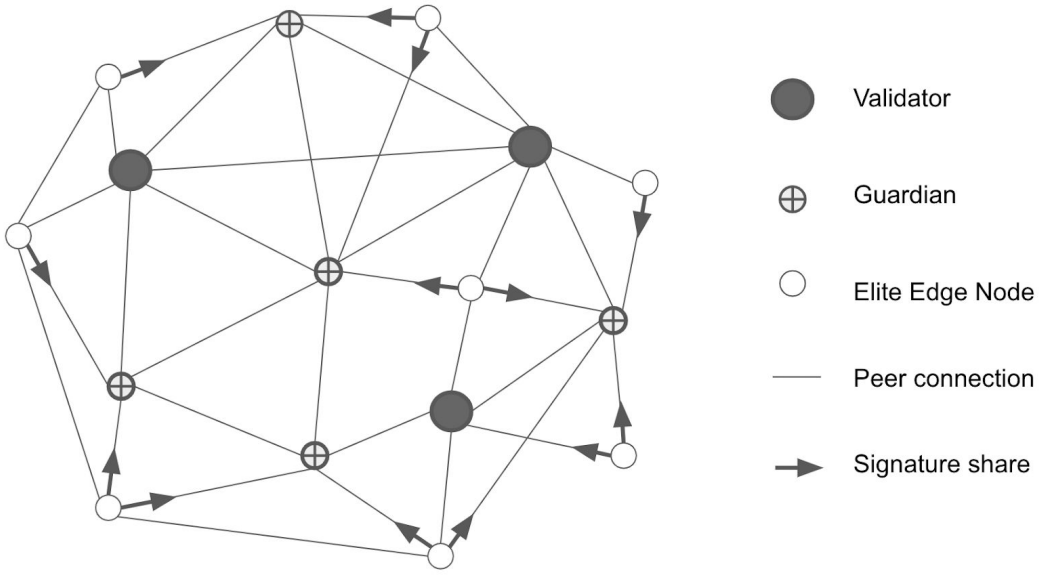

Diagram 5. The joint mesh network consisting of the Validators, Guardians, and Edge Nodes. The Validators propose and finalize new blocks. The Guardians seal the blockchain by collectively signing the checkpoint blocks. The Edge Nodes are responsible for data delivery and providing other useful services. To prove its uptime, an Elite Edge Node broadcasts its BLS signature share of the checkpoint blocks to the Guardians it is connected to. Then, through the aggregated signature gossip routine, the BLS signature shares of the active elite nodes are aggregated by the Guardians into one signature, which then gets written into the blockchain through the new blocks proposed by the validators. An elite node is eligible to split the TFuel inflation rewards if the aggregated signature recorded on-chain contains its BLS signature share.

The Theta blockchain will add the following interface to support TFuel staking for Elite Edge Nodes. To stake TFuel to an elite node, the *source* wallet needs to sign the ***DepositStakeToEdgeNode*** transaction with the BLS summary of the edge node (similar to guardian nodes).

$$\textbf{\textit{DepositStakeToEdgeNode}}(edgeNodeSummary, source) \quad (1)$$

An elite node can withdraw the TFuel stake by issuing a ***WithdrawStakeFromEdgeNode*** transaction as defined below any time. After the withdrawal transaction is confirmed on the blockchain, there is a pending withdrawal phase (28800 blocks) during which the elite node does not earn TFuel inflation rewards.

$$\textbf{\textit{WithdrawStakeFromEdgeNode}}(edgeNodeAddress) \quad (2)$$

By default, all the Uptime Mining rewards go to the *source* wallet. To facilitate delegated TFuel staking, the Theta blockchain will provide an additional interface for the elite node to split the reward:

$$\textbf{\textit{SplitEdgeNodeReward}}(\textit{beneficiaries}, \textit{splits}) \quad (3)$$

This interface allows an elite node to sign a transaction to specify 1) the *beneficiaries*, a list of wallets to split the Uptime Mining TFuel rewards in addition to the *source* wallet, and 2) the *splits*, the split percentage of each beneficiary.

Note that Uptime Mining does not tie to the “performance” of the elite node, i.e. the amount of data it relays. This is intended since we believe availability is the first step towards usability and thus should be rewarded. Further, with the launch of the edge compute feature, Edge Node extends its functionality to other other areas such as generic computing. Uptime and availability is the common stepping stone for all these use cases and has its own importance.

Later in the whitepaper we will introduce the Proof-of-Relay mechanism which motivates the Edge Nodes to not just simply be available but also perform useful work for platform partners.

4. TFuel Burning

The Service Quality Feedback Loop

Traditionally, video streaming platforms like Youtube and Twitch pay a centralized content delivery network (Akamai, AWS Cloudfront, etc) for stream delivery services. A platform might be able to use the delivery network for free but the service would be limited and likely without quality guarantee. To receive better service guarantees, a platform needs to pay for the delivery service. Theta Network will follow a similar business model, but process service payments in a fully decentralized and transparent fashion.

As in traditional content delivery services, platforms need to pay Theta network in order to use the infrastructure with service guarantees. However, on the Theta Network, these payments are in TFuel and recorded on-chain. Based on the recent on-chain platform payment records, each individual edge node independently determines the **priorities** of the platforms’ traffic. Higher payers get higher priority. This way, larger platforms can receive stable and reliable services by paying market competitive fees. Individual users (i.e. freelance streamers) can still use the Edge Network for free, but without service quality guarantees. If there is more demand from platforms than the bandwidth supplied by the edge network, then network fees (ie. the amount of TFuel burned) will naturally increase similar to gas price of Ethereum.

Additionally, by leveraging the blockchain ledger, a platform can manage payments at a much finer granularity. For example, a platform can set different amounts of payment for different streamers, depending on each streamer's viewership or status. Today's video platforms typically require streamers to register themselves and in turn this can be recorded on the blockchain. In this way, a platform can also deregister streamers if they behave maliciously.

Diagram 6 below illustrates the **service quality feedback loop** discussed above. Higher TFuel payment amounts for a streamer leads to higher priority for that streamer's traffic, and hence improved delivery quality, and vice versa. **A portion of the TFuel payments to the network will be burned permanently**, and the remainder will be split among the elite nodes that submitted a Proof-of-Relay. Initially, the protocol requires that at least 25% of each payment will be burned.

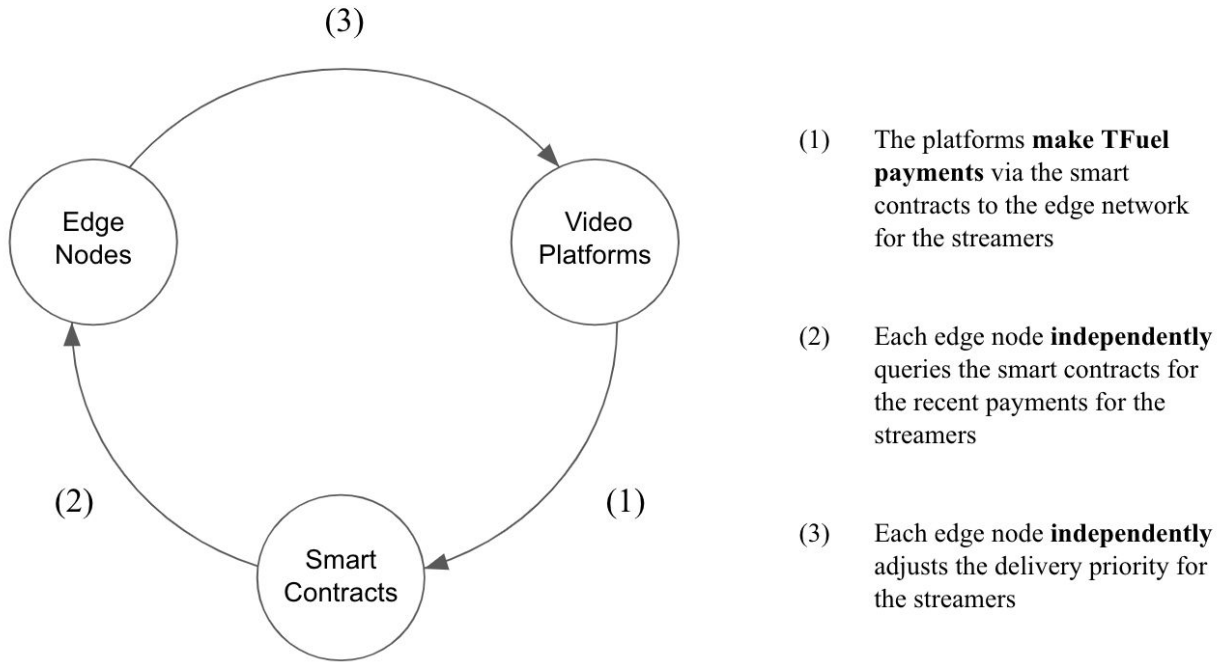

Diagram 6. The service quality feedback loop among the three key parties, the Edge Nodes, the video platforms, and the smart contracts hosted on the blockchain.

An Elite Edge Node can query the Guardian Nodes it connected to for the recent payments recorded on-chain. Then, it can **prioritize** a streamer's traffic based on the sum of recent platform payments for the streamer, and its own assessment of how popular the content is, e.g. $priority = paymentSum / estimatedTotalViewer$. The total viewer count can be estimated by stats like how many downstream nodes the Edge Node connects to.

Through this incentive system, we can achieve the following **goals**:

- We want to encourage honest behavior. A platform should pay (and therefore burn) a sufficient amount of TFuel for using the Theta Edge network infrastructure with service

guarantees, which provides incentives for the Edge Nodes to provide high quality data delivery service.

- Byzantine Edge node behavior resistance. Even if a large fraction of the edge nodes (e.g. 60%) deviates from the prescribed protocol arbitrarily, the delivery network should still function properly, and a significant amount of TFuel still gets burned to effectively reduce the amount of circulated TFuel.

- The system needs to be able to prevent malicious attempts from streamers. For example, the “identity theft” attack where a stream pretends to be another streamer. And the “free riding” attack, i.e., tricking the system so the edge nodes consider the streamer should be paid for by a platform that they didn’t register with.

The “Attack Vector Analysis” section will explain how these goals can be achieved under reasonable assumptions.

Table 1. Comparison of the traditional and the Theta approach

Platform/Streamer relationship management	Traditional approach	A streamer creates an account in the centralized database of the platform
	Theta Network	The platform calls an on-chain smart contract to register the streamer for the platform. After that, anyone can query the blockchain to verify that the streamer is registered on the platform.
Usage stats	Traditional approach	Cloud service collects and provides usage stats. No proof of usage is provided.
	Theta Network	Each individual edge node collects and evaluates the usage of each platform. In particular, for each video stream, the content provider/streamer needs to provide metadata including (platform, streamer). They also need to sign the metadata. As an edge node relays the stream, it examines and verifies the associated metadata. If the signature is valid, the edge node would query the blockchain to see if the platform has registered the streamer. If both conditions are met, the edge node can be convinced that the stream data is from the claimed streamer and is delivered for the claimed platform.
Payment	Traditional approach	Cloud service sends invoice to the platform based on usage, and the platform pays monthly
	Theta Network	The platform pays through smart contracts on-demand. The payment records for (platform, streamer) are available and provable on-chain.
Service	Traditional approach	Cloud service determine which services the platform can use based on the contract/payment

	Theta Network	Each individual edge node determines the priority of a content based on the usage and payment from the platform for the content.

Implementation Remarks

We plan to implement the above described incentive system **at the smart contract level** for maximum flexibility, extendability and decentralization. Should there be any necessary rule changes, a new smart contract can simply be re-deployed with the upgraded rules, and Edge Nodes can point to the new contract. This also allows us to extend to other use cases in the future by deploying new smart contracts running in parallel with the contracts governing the incentive mechanism for the video streaming use case. This opens up a whole of possible applications beyond video. To implement TFuel burning in the smart contract level, the smart contract can simply send the TFuel being burned to address 0x0. This makes those TFuel non-recoverable and effectively takes them out of circulation. Below we define the interface for the smart contract governing the platform payment.

```
interface PlatformPaymentProcessor {

    // Function to be called by a platform to register a streamer, where msg.sender
    // is the wallet address of the platform
    function registerStreamer(address streamer) public returns (bool);

    // Function to be called by a platform to deregister a streamer, where msg.sender
    // is the wallet address of the platform.
    function deregisterStreamer(address streamer) public returns (bool);

    // Function to be called by an Elite Edge Node to submit the proof-of-relay, where
    // msg.sender is the wallet address of the Elite Edge Node. The function should verify
    // the proof and record the verification results.
    function submitProofOfRelay(address platform, address streamer,
         bytes memory proofOfRelay) public;

    // Function to be called by a platform to make network payment, where msg.sender is the
    // platform making the payment, and msg.value represents the TFuel payment sent by the
    // platform. The function should burn a portion of the payment, and split the remainder
    // based on the proof-of-relay verification records.
    function payToNetwork(address streamer) public payable returns (bool);

    // Function to return the total amount of TFuel paid by a platform for a streamer
    // between the two block heights. The payment sum can be used by each individual Elite Edge
    // Node to determine the priority of the traffic for the streamer.
    function getNetworkPaymentSum(address platform, address streamer,
         uint256 startBlockHeight) public view returns (uint256 paymentSum);
}
```

The two functions `registerStreamer` and `deregisterStreamer` should be called by the video platforms to register/deregister a streamer. The smart contract implementing the above interface should keep track of the relationship between the platforms and the streamers. The function `submitProofOfRelay` is intended for the elite nodes to submit the Proof-of-Relays.

The function should verify the submitted proof. If the proof passes the validation, the function should record that the Edge Node submitted a valid Proof-of-Relay for the target streamer. The function `payToNetwork` is for the platforms to submit TFuel payments to the network. As prescribed by the protocol, a portion of the payment will be burned, and the remainder should be split based on the valid Proof-of-Relay records since the last payment. Finally, the function `getNetworkPaymentSum` can be called by the Edge Nodes. It returns the total amount of TFuel paid by a platform for a streamer between the two specified block heights. The payment sum can be used by each individual elite node to determine the priority of the traffic for the streamer as discussed earlier.

Attack Vector Analysis

Byzantine Edge node behavior resistance. If an attacker controls a portion of the Edge Nodes, those Edge Nodes could become byzantine nodes and behave arbitrarily. For example, the byzantine Edge Nodes might not rank the platform traffics based on the recent payments. However, X% of byzantine Edge Nodes only reduces the delivery service efficiency by X%, e.g. even if 60% of the Edge Nodes are byzantine nodes, the edge network can still provide 40% of the total bandwidth capacity. The platforms might pay less due to the service quality drop, but these payments still lead to TFuel burning. Therefore, albeit slower, eventually the system can still reach the balanced state as discussed in the “Enhanced Network Economics” section.

“Identity theft” attack and mitigation: “Identity theft” refers to an attack where a streamer pretends to be another streamer by including bogus information in the metadata sent along with the video stream. This can be prevented in our proposed system if we require the streamer to sign the video packets he/she sends out to the network. The signature should sign the hash the data packet concatenated with the platform address, a random nonce, and the current block height:

$$streamer\_signature(video\_data\_hash \,||\, platform \,||\, rand\_nonce \,||\, block\_height) \qquad (4)$$

Any Edge Node receiving a video packet and the accompanying signature can verify its validity, hence the identify theft attack is not possible.

“Free riding” attack and mitigation: “Free riding” refers to tactics where streamers specify a platform they are not currently registered with, hoping that the platform will handle the TFuel payments for them.

To counteract such malicious attempts, each video data packet from a streamer should contain the following information: i) the platform address, ii) the streamer address, and iii) the signature as specified in Formula (4) above.

This way, an Edge Node can verify that 1) the packet does come from the claimed streamer based on the signature verification, and 2) with the platform and streamer address, the Edge

Node can look up the blockchain to get the latest payment record of the platform for the streamer to determine the priority of the video data packet.

If a streamer does not currently register with a platform, then the blockchain will not record the association. So, if streamers specify a platform they are not currently associated with, after querying the blockchain, an Edge Node can easily detect the malicious behavior and discard the video packet. This effectively mitigates the “free-riding” attack.

5. Proof-of-Relay

Performance-Based Reward via Proof-of-Relay

To receive the performance-based rewards, an Edge Node first needs to stake a sufficient amount of TFuel to upgrade itself to an **Elite Edge Node**. TFuel staking is necessary in order to mitigate sybil attacks (see the “Attack Vector Analysis” section).

An Elite Edge Node can gain performance-based reward by submitting the **“Proof-of-Relay”** to the blockchain. This is to encourage edge nodes by rewarding them based on how much data they actually relay. Denote the hash of the video packet sent by a streamer by $video\_data\_hash$, and the signature of the streamer defined in Formula (4) by $streamer\_signature$. The Proof-of-Relay is composed of two checks:

$$\textbf{Condition 1: } HASH(streamer\_signature \,||\, platform \,||\, streamer \,||\, edge\_node) < M \tag{5}$$

$$\textbf{Condition 2: } block\_height > block\_height\_when\_submitted - H \tag{6}$$

Obviously, if an edge node relays more data for a registered streamer, it would have a higher probability to meet Condition 1. A careful reader might wonder if it is possible for a malicious Elite Edge Node to produce bogus Proof-of-Relays. In the “Attack Vector Analysis” part later in this section, we will argue that under reasonable and practical assumptions, an elite node should not be able to cheat and gain much by forging fake Proof-of-Relays.

Condition 2 ensures that an edge node cannot use a streamer signature in the remote past to generate the hash. H should be set to a sufficiently small number (comparable to the duration of a video segment, much smaller than the TFuel pending withdrawal period). Without this condition, a new elite node might attempt to test all the streamer signatures sent in the past to see if any signature satisfies Condition 1. This gives it an unfair advantage over other honest elite nodes. Accounting for recency addresses such issues.

Whenever obtaining a valid Proof-of-Relay, the edge node can submit it to blockchain (via a smart contract). Later when the platform pays TFuel, as discussed in the previous section, part

of the TFuel is burned, and the remainder TFuel will be rewarded to the edge nodes that submitted the Proof-of-Relay.

Attack Vector Analysis

Leech Edge Node: A malicious edge node might only compute the hash as defined in Formula (5) without relaying the data to the downstream nodes. First of all, if only a portion of the edge nodes behave this way, the delivery network consisting of the remaining honest nodes can still function (although the efficiency might decrease a bit). Furthermore, we can implement the BitTorrent “tit-for-tat”-like protocol to encourage the sharing behavior. With such protocols in place, in the long run the leech nodes would have less chance to obtain data from other nodes, and thus its probability of generating a relay proof becomes lower. Hence, an economically rational edge node has the incentive to honestly relay the video data.

Elite Edge Node sybil attack: A malicious Elite Edge Node operator might instantiate many edge node identities. Those nodes do not relay data, but only compute the hash as defined in Formula (5). Such a sybil attack could be costly to launch. To see why this is the case, we first note that since the address of an elite node cannot change, the edge node operator has no way to alter the string being hashed (see Condition 1) for that elite node. Thus, unlike a Bitcoin miner, an elite node cannot simply try arbitrary strings to make the hash lower than the threshold M. To “try” different strings, the malicious operator would need to unbind the TFuel staked to an elite node, and stake to another node. With this tactic, potentially the one TFuel token can be used multiple times to generate different hashes. However, TFuel staking has a pending withdrawal period of 28800 blocks. Thus, the operator cannot shuffle its TFuel among its edge nodes frequently enough to gain advantages. Hence, to launch a sybil attack, the malicious operator needs to stake TFuel for a large number of nodes, making the sybil attack costly. Furthermore, as pointed out in the analysis for leech nodes, the operator is better off economically if the nodes under its control actually relay data.

Malicious streamer: There is a caveat in the above sybil attack analysis, in that if a streamer colluded with the edge nodes, the streamer might generate a large number of video data and the corresponding signature in a short amount of time, or bias the video data to increase the probability that the colluded edge node can generate valid relay proofs. However, the platform can easily detect such abnormalities statistically. For example, the platform can record the live stream of the streamer, and check if the video data hash is genuinely from the recorded stream. If a malicious attempt from a streamer is detected, the platform can simply deregister the streamer. The platforms have incentive to monitor the adversarial behaviors since the performance-based TFuel reward comes out of their pocket.

Malicious platform: A malicious platform might skip payments. As a result, some of the Edge Nodes might end up delivering some data for the platform without being paid. However, if an Edge Node checks the recent on-chain payment records of the platform sufficiently frequently, it

can detect the delinquency in time and stops serving the platform. Thus, the exposure of the Edge Nodes can be effectively limited.

6. Conclusions and Future Extensions

In this whitepaper, we presented the protocol enhancements and crypto economics design for Theta Mainnet 3.0. In particular, we introduced a new role called **Elite Edge Node** which can earn TFuel rewards through **Uptime Mining** and **Proof-of-Relay**. A regular edge node can stake TFuel to become an elite node. We also introduced the concept of **TFuel burning** as the cost for platforms to use the Theta edge network for data delivery. Acting as a balancing force to TFuel inflation, TFuel burning can effectively adjust the circulation supply of TFuel to fit the actual demand and maximize the utility value of TFuel and ultimately the utility value of Theta staked against validators and guardians.

More generally, the Theta edge network can work not only as a data delivery network, but also as a generic edge computing platform [5, 6]. Such a platform allows Task Initiators to post tasks for edge nodes to download and solve. Task initiators also register the tasks and provide the TFuel rewards for each task on the blockchain through smart contracts. Tasks can be anything ranging from solving a set of equations, finding novel protein structures to help fight COVID-19, transcoding a video, to thousands of other applications that can leverage a network of distributed edge computing devices.

Edge Nodes can work as a generic computational platform which can host various software including the solver for the tasks issued by task initiators. Edge nodes poll the task initiators to download tasks. Once a task is solved, the edge node can upload the solution to relevant **Smart Contracts** on the blockchain. These smart contracts act as the verifier for the solution, and as a trustless escrow for the task rewards. Once a submitted solution is verified, smart contracts will transfer the reward to the solver (i.e. a particular edge node) automatically and transparently. If a Task Initiator does not want to reveal the solutions on-chain, **zero-knowledge proof techniques** (e.g. zk-SNARK) can be leveraged. Once an edge node solves a task, it can encrypt the solution using the task initiator's public key, and submit the encrypted solution to the smart contracts.

Meanwhile, an edge node can also generate a zero-knowledge proof showing that the submitted solution is encrypted using the task initiator's public key, and the plaintext solution indeed solves the corresponding task. Smart contracts simply verify the proof and reward the edge node if the proof passed the check. On the other hand, the task Initiator can download the encrypted solution from the blockchain and decrypt it using its own private key. Such a decentralized edge computing framework eliminates all counterparty risks and is thus able to effectively incentivize the edge nodes to share their unused computing resources.

Theta crypto economics can naturally be extended to handle any generic computing. For example, the zero-knowledge proof for the edge compute tasks is similar to the Proof-of-Relay for the data delivery tasks. Thus, the smart contracts which handle the Proof-of-Relay rewards can be generalized to manage rewards for generic computing tasks. Similar to data delivery, smart contracts can enforce a rule that **a certain percentage of the escrowed TFuel reward should be burned** as the cost for using the Theta edge computing infrastructure. With the edge network expanding its capability beyond data delivery to cover generic edge computing, we believe the utility value of the network and TFuel can be extended substantially in the future.

References

[1] Long, J., & Wei, R. (n.d.). Scalable BFT consensus mechanism through aggregated signature gossip. *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. https://arxiv.org/pdf/1911.04698.pdf

[2] Theta Labs, Theta Blockchain Whitepaper. https://s3.us-east-2.amazonaws.com/assets.thetatoken.org/Theta-white-paper-latest.pdf

[3] Theta Labs, Introducing Theta Edgecast: The world's first decentralized streaming DApp for end-to-end live streaming, transcoding, caching and video delivery. https://medium.com/theta-network/introducing-theta-edgecast-the-worlds-first-decentralized-streaming-dapp-for-end-to-end-live-e08c875a7f86

[4] Bordo, M, Equation of Exchange, *1989 Money*. https://link.springer.com/chapter/10.1007/978-1-349-19804-7_17

[5] Theta Labs, Theta Decentralized Edge Computing Platform. https://s3.us-east-2.amazonaws.com/assets.thetatoken.org/Theta-Decentralized-Edge-Computing-Platform.pdf

[6] Theta Labs, Theta Network Introduces Edge Compute aiding Folding@home's fight against COVID-19 and other diseases. https://medium.com/theta-network/theta-network-introduces-edge-compute-aiding-folding-homes-fight-against-covid-19-and-other-aac8742aeb12

[7] EIP 1559: The Final Puzzle-Piece to Ethereum's Monetary Policy https://medium.com/@TrustlessState/eip-1559-the-final-puzzle-piece-to-ethereums-monetary-policy-58802ab28a27