WAVES

Dynamic Transaction Fee and Fee-Burn Monetary Policy Design

Node Revenue, WAVES Burn Mechanism, and Critical Infrastructure Protection

Author: @Cincirman

August 26, 2026

A technical-economic proposal for community-governed transaction fees, fee burn, generator economics, and protected infrastructure on Waves.

Contents

Executive Summary

The current transaction fee structure on the Waves network has largely lost its economic significance, particularly during periods when the WAVES price is low. Minimum fees of 0.001 WAVES for Transfer, 0.003 WAVES for Exchange, and 0.005 WAVES for a basic Invoke Script transaction translate into costs that are almost negligible in USD terms. Waves documentation confirms these base minimums, with additional fees applying in cases such as smart accounts, smart assets, and certain Invoke actions.

Using the approximately $0.21645/WAVES price contained in the WavesOnChain dataset for August 26, 2026, the current and proposed fee levels can be expressed as follows:

Transaction	x1	x1 USD	x10	x10 USD	x20	x20 USD
Transfer	0.001	$0.00022	0.010	$0.00216	0.020	$0.00433
Exchange	0.003	$0.00065	0.030	$0.00649	0.060	$0.01299
Invoke Script	0.005	$0.00108	0.050	$0.01082	0.100	$0.02165
Data, 1 KB	0.001	$0.00022	0.010	$0.00216	0.020	$0.00433
Data, 4 KB	0.004	$0.00087	0.040	$0.00866	0.080	$0.01732

The table makes the core argument of this proposal clear. Even a 20x increase in Waves transaction fees would place a standard Transfer transaction at only about 0.43 cents, while a basic Invoke would cost approximately 2.16 cents at the current WAVES price.

> **In other words, fees can become economically meaningful without turning Waves into a high-fee blockchain.**

The model proposed in this paper introduces three fee regimes:

$$M \in \{1, 10, 20\}$$

Generator nodes would vote on the active multiplier in a manner similar to the existing Waves Community-Driven Monetary Policy mechanism.

The feature that introduces the system would first need to pass the normal Waves Feature Activation process, requiring 80% support over 10,000 blocks on Mainnet. Once activated, subsequent fee multiplier changes would operate as a monetary-policy parameter and require a 5,001 out of 10,000 block majority, following the same absolute-majority principle used by the current block reward voting mechanism.

The proposed economic structure is:

Fee level	Burn	Generator pool
x1	0%	100%
x10	50%	50%
x20	50%	50%

The x1 regime should not apply a burn. It should function as both the low-fee regime and the network's economic safety mode. If market conditions no longer justify higher transaction costs, generators can return the network to its current fee economics without simultaneously reducing node income through fee burning.

At x10 and x20, half of every network fee would be permanently removed from the WAVES supply and the remaining half would be distributed to generators.

If the existing Waves-NG 40% current generator / 60% next generator fee-sharing mechanism is preserved, x10 and x20 fees would effectively be distributed as:

50% Burn + 20% Current Generator + 30% Next Generator

Assuming transaction volume remains unchanged, this means x10 increases aggregate node transaction-fee revenue by 5x, while x20 increases it by 10x, even though half of every fee is burned.

The WavesOnChain Network Daily Total Fee dataset allows the economic impact of this model to be quantified using actual network activity.

30-day average daily fee = 237.997 WAVES/day

For the 30 completed days between July 27 and August 25, 2026, the median was approximately 238.828 WAVES/day, while observed daily values ranged from approximately 213.84 WAVES to 293.71 WAVES.

Regime	Total fees/day	Burn/day	Generator fees/day	Relative node fee revenue
x1	237.997	0	237.997	1x
x10	2,379.967	1,189.983	1,189.983	5x
x20	4,759.934	2,379.967	2,379.967	10x

At x20, the theoretical annualized burn from transaction fees alone would be approximately 868,688 WAVES/year under constant activity.

This simultaneously addresses two economic weaknesses in the current network structure: transaction-fee income becomes meaningful for generators, while real network usage begins creating a continuous WAVES supply sink.

A separate issue appears at the infrastructure layer. Waves is not composed only of end-user transactions. Systems such as Unit0 produce high-frequency L0 Invoke transactions as part of their normal protocol operation. Unit0 depends on the Waves Chain Contract to record block metadata and coordinate its relationship with Waves consensus.

For that reason, I propose that Unit0's consensus-critical transactions remain outside the global x10/x20 multiplier and continue operating at the x1 fee level. Because x1 also carries no burn under this design, the existing fee economics of these Unit0 transactions are preserved.

> **This allows a single upgrade to address three separate economic objectives: introducing a WAVES burn mechanism, materially increasing node fee revenue, and preventing an additional cost shock for Unit0 nodes whose high-frequency L0 operations already require a more sustainable fee structure.**

1. Current Transaction Fee Structure on Waves

The Waves transaction fee model is based primarily on transaction-type-specific minimum fees rather than a continuously auctioned gas market.

Some of the main base values are:

Transaction	Minimum fee

Transfer	0.001 WAVES
Exchange	0.003 WAVES
Invoke Script	0.005 WAVES + additional actions
Data	0.001 WAVES / KB
Burn	0.001 WAVES
Create Alias	0.001 WAVES
Lease	0.001 WAVES
Lease Cancel	0.001 WAVES
Reissue	0.001 WAVES
Sponsor Fee	0.001 WAVES
Update Asset Info	0.001 WAVES
NFT Issue	0.001 WAVES
Regular Issue	1 WAVES
Set Asset Script	1 WAVES
Mass Transfer	0.001 + 0.0005 x N
Ethereum-like Transfer	Transfer fee
Ethereum-like Invoke	Invoke fee

Smart accounts and smart assets can add approximately 0.004 WAVES to some transaction types. For example, an Exchange transaction involving one smart asset may require 0.007 WAVES, while two smart assets can raise the requirement to 0.011 WAVES. Invoke transactions begin at 0.005 WAVES, while actions such as non-NFT asset issuance can add further fees.

It is therefore important not to destroy the relative relationship between transaction complexity and cost.

One of the important objections raised during the 2019 New Fee Calculation Model discussion was that resource-intensive transactions should not simply be priced identically to simple transactions.

The model proposed here avoids that issue.

Instead of redesigning the entire fee curve, it preserves the existing relative fee structure and moves the entire curve up or down using a governance-controlled multiplier:

$$\textbf{Required Fee} = \textbf{Current Fee} \times M$$

$$M \in \{1, 10, 20\}$$

This is much simpler than replacing the existing fee architecture with an entirely new resource-pricing model.

2. Why x1 / x10 / x20?

A continuously changing WAVES/USD-based consensus fee might initially appear attractive. In practice, however, tying minimum valid transaction fees directly to an external price oracle would introduce unnecessary consensus risks.

If every node needs an external USD price in order to determine whether a transaction is valid, new failure modes appear:

- oracle manipulation
- oracle outages
- disagreement between price sources
- stale data
- consensus divergence
- potential chain forks

For that reason, the USD price should not become part of consensus.

$$M \in \{1, 10, 20\}$$

Generator operators can evaluate external market conditions themselves and vote for the multiplier they consider economically appropriate.

The community can still publish a non-binding USD fee corridor as guidance.

For example, suppose the soft targets are:

- Transfer <= $0.01
- Invoke <= $0.05

Under x20:

$$0.020P \leq 0.01$$

$$0.100P \leq 0.05$$

$$P \leq \$0.50$$

Therefore, a practical policy guide could look like:

WAVES price zone	Reasonable fee regime
~$0-$0.50	x20
~$0.50-$1.00	x10
~$1+	x10 or x1
Significantly higher WAVES price	x1

These values would not be consensus rules. They would simply provide generators with an economic reference when deciding how to vote.

At the August 26, 2026 price of approximately $0.21645, x20 sits comfortably inside such a corridor.

3. Fee Voting Mechanism

Two different governance processes need to remain clearly separated.

A. DFMP Feature Activation

The Dynamic Fee Monetary Policy feature itself should first pass the normal Waves Feature Activation Protocol.

On Mainnet:

- voting period: 10,000 blocks
- approval threshold: 8,000 blocks
- support required: 80%
- followed by the normal activation period

This ensures that introducing a new consensus-level fee mechanism still requires broad network agreement.

B. Multiplier Monetary Policy

Once DFMP is activated, moving between x1, x10, and x20 should not require a new feature vote every time.

Instead, the active fee multiplier becomes a monetary-policy parameter, similar to the block reward.

Votes(x1), Votes(x10), Votes(x20)

Votes(M) >= 5,001

For example:

- x20: 4,700
- x10: 3,500
- x1: 1,800

This does not result in x20 activation. x20 has the largest number of votes, but it has not reached the required absolute majority of 5,001 blocks.

If no option reaches the threshold, the current multiplier remains unchanged. This avoids unstable policy changes based on simple plurality.

4. The Special Role of x1: Safety Regime

M = 1 -> Burn = 0

Generator Share = 100%

There are two reasons for this.

First, if the WAVES price rises enough for the community to reduce fees back to x1, there is little reason to simultaneously cut node income through fee burning.

Second, x1 becomes a genuine economic fallback mode.

If unexpected issues appear after the reform, generators can vote the network back to:

- the current minimum fee levels
- the current fee distribution structure
- zero fee burn

This makes the reform reversible at the monetary-policy level rather than turning it into a permanent fee shock that requires another protocol upgrade to undo.

5. 50% Burn / 50% Generator at x10 and x20

Burn = 50%

Generator Pool = 50%

The general concept already has precedent in blockchain fee economics. Solana, for example, has used a fee structure in which part of the base transaction fee is burned and part is distributed to validators.

The Waves implementation would differ in one important way: the total nominal fee level would not be determined by a continuously changing block-demand market. It would instead be determined by a generator-voted monetary-policy multiplier.

Preserving Waves-NG

Waves-NG currently distributes transaction fees approximately as 40% to the current generator and 60% to the next generator. This incentive structure should be preserved.

Generator Pool = 0.50f

Current Generator = 0.40 x 0.50f = 0.20f

Next Generator = 0.60 x 0.50f = 0.30f

50% Burn + 20% Current Generator + 30% Next Generator

Consider a basic Invoke transaction under x20:

0.005 x 20 = 0.100 WAVES

Destination	WAVES
Burn	0.050
Current generator	0.020
Next generator	0.030
Generator total	0.050

Today, the generator sector eventually receives approximately 0.005 WAVES from the same basic transaction. Under x20:

0.050 / 0.005 = 10

Aggregate generator transaction-fee revenue therefore rises by 10x per equivalent transaction.

6. Burn and Node Economics Using WavesOnChain Data

The most useful baseline for this proposal is the WavesOnChain Network Daily Total Fee dataset.

For the 30 completed days from July 27 through August 25, 2026:

F30 = 237.9967 WAVES/day

The median is approximately 238.8281 WAVES/day, with an observed minimum of approximately 213.8394 WAVES/day and an observed maximum of approximately 293.7080 WAVES/day during the same period.

Because these are realized network fees rather than hypothetical transaction assumptions, they provide a strong baseline for evaluating the reform.

6.1 Constant-Activity Scenario

x1: Burn = 0; Generator Revenue = F

x10: Burn = 5F; Generator Revenue = 5F

x20: Burn = 10F; Generator Revenue = 10F

Regime	Users pay/day	Burn/day	Generator pool/day
x1	237.997	0	237.997
x10	2,379.967	1,189.983	1,189.983
x20	4,759.934	2,379.967	2,379.967

Regime	Estimated annual burn
x1	0 WAVES
x10	434,344 WAVES
x20	868,688 WAVES

These figures assume unchanged transaction activity and transaction composition. They should therefore be treated as baseline scenarios, not guaranteed forecasts.

7. What If Transaction Demand Falls?

It would be unrealistic to assume that increasing transaction fees has no effect on demand.

This was also one of the strongest criticisms raised during the 2019 WEP-5 discussion: increasing fees by 10x does not automatically mean fee revenue increases by 10x because some transaction activity may disappear.

Define the new fee-generating activity ratio as:

$$q = Activity\_new / Activity\_old$$

For x10 and x20:

$$Generator\ Revenue = 0.50 \times M \times q \times F$$

$$Burn = 0.50 \times M \times q \times F$$

Activity retained	Regime	Burn/day	Node fee income/day	Relative to today
100%	x10	1,189.98	1,189.98	5.0x
75%	x10	892.49	892.49	3.75x
50%	x10	594.99	594.99	2.5x
25%	x10	297.50	297.50	1.25x
100%	x20	2,379.97	2,379.97	10x
75%	x20	1,784.98	1,784.98	7.5x
50%	x20	1,189.98	1,189.98	5x

25%	x20	594.99	594.99	2.5x

Generator fee revenue reaches break-even when:

$$0.50 \times M \times q = 1$$

For x10, q = 0.20. This means fee-generating activity could decline by as much as 80% before aggregate node transaction-fee income falls below its current level.

For x20, q = 0.10. In theory, activity could fall by as much as 90% before aggregate node transaction-fee revenue falls below today's baseline.

This gives x20 a substantial demand-elasticity buffer from the perspective of node economics.

8. Does the Burn Make WAVES Deflationary?

Transaction fee burn and net token deflation are not the same thing.

$$\text{Net Issuance} = \text{Block Rewards} - \text{Fee Burn}$$

Therefore, the strongest claim that can be made without combining all monetary variables is:

DFMP introduces a permanent activity-linked deflationary pressure on the WAVES supply.

Whether the total WAVES supply actually contracts over a given period depends on:

- the active block reward
- number of generated blocks
- transaction activity
- active fee multiplier
- share of protected transactions

The proposal does not need WAVES to become immediately net-deflationary in order to be economically useful.

Its more important property is that real network usage begins removing WAVES from supply instead of transaction fees flowing almost entirely back into the generator economy.

9. Unit0: A Second Problem the Fee Reform Can Address

Unit0 should not be treated as an ordinary dApp.

Unit0 operates as a network connected to Waves and relies on a Chain Contract on Waves for parts of its consensus workflow. Generator activity includes interactions that write Unit0 block information to the Waves chain and use the Chain Contract as part of the coordination between Unit0 and Waves consensus.

These transactions therefore have a different economic character from ordinary user transactions.

9.1 Why Unit0 Should Remain Outside the Multiplier

Unit0 nodes already produce a very high number of Waves Invoke transactions.

Using an observed short-term traffic example of approximately 285 Invoke / 20 minutes, a simple 24-hour linear extrapolation produces:

$$285 \times 72 = 20{,}520 \text{ Invoke/day}$$

This should not be interpreted as a long-term Unit0 network average. It is only an illustration of how high-frequency infrastructure behaves under a multiplier.

$$20{,}520 \times 0.005 = 102.6 \text{ WAVES/day}$$

Regime	Illustrative Unit0 gross fee/day
x1	102.6 WAVES
x10	1,026 WAVES
x20	2,052 WAVES

This shows why the same x20 fee that represents only a few cents for a normal user can create a fundamentally different economic impact for machine-generated consensus activity.

I do not think this is necessary.

9.2 Proposed Unit0 Treatment

Effective Multiplier = 1

For consensus-critical Unit0 Chain Contract calls, this should apply even when the global multiplier is x10 or x20.

Effective Multiplier = 1 -> Burn = 0

This does not create a new subsidy for Unit0. It does not lower the current fee below today's level. It simply prevents a general Waves fee reform from increasing an already significant infrastructure cost by 10x or 20x.

The fee upgrade therefore becomes an opportunity to solve several related issues in one protocol change:

1. normal Waves transaction fees become economically meaningful;
2. WAVES gains an activity-linked burn mechanism;
3. Unit0 generators are protected from an additional high-frequency L0 cost shock.

10. Protected Infrastructure Design

Unit0 protection should not be implemented as a simple sender-address whitelist.

Node addresses can change, and address-based exemptions can eventually create governance rent-seeking.

A cleaner model is:

Effective Multiplier = min(Global Multiplier, Cap(dApp, function))

Infrastructure	Fee policy
Unit0 consensus calls	x1 / no burn
Critical price oracle / keeper	maximum x10
Normal dApps	global x1/x10/x20
Normal users	global x1/x10/x20

Protected status should preferably be defined using dApp address + callable function rather than sender identity.

For example, only the consensus-critical functions of the Unit0 Chain Contract could receive x1 treatment. Other unrelated economic functions in the same contract would not automatically qualify.

Changes to the protected registry should also not use the ordinary 50% monetary-policy threshold. Granting a fee advantage to a contract is different from changing the network-wide fee level.

A higher threshold - potentially the same 80% feature-style support used for consensus features - would be safer.

11. Price Oracles and High-Frequency Smart Contracts

Price oracles are not necessarily as consensus-critical as Unit0, but they can be essential for DeFi stability.

Oracle updates can directly affect:

- collateral valuation
- liquidations
- peg stabilization
- lending
- synthetic asset pricing
- arbitrage

Using approximately 26 Invoke transactions per hour as an operational scenario:

26 x 24 = 624 Invoke/day

624 x 0.005 = 3.12 WAVES/day

Regime	WAVES/day	Approx. USD/day	WAVES/30 days
x1	3.12	~$0.68	93.6
x10	31.2	~$6.75	936
x20	62.4	~$13.51	1,872

A single oracle may be able to absorb x20 economically. The wider problem appears when many oracle, keeper, liquidation, and stabilization services operate continuously.

More importantly, an operator may respond to higher fees by lowering update frequency. That could result in:

- more stale price data
- delayed liquidations
- wider arbitrage bands
- reduced stablecoin stabilization efficiency

Recommended starting cap for approved oracle/keeper infrastructure: x10

During a global x20 period, a normal Invoke would cost 0.100 WAVES while a protected oracle Invoke would cost 0.050 WAVES. This still raises infrastructure fees relative to today, but avoids the full x20 increase.

12. Would Smart Contract Fee Accounting Become Confusing?

Network transaction fees and application-level payments are separate concepts.

An Invoke Script transaction may contain a payment to a dApp, but its network fee is a separate amount charged to the transaction sender.

Therefore, dApp business fees, payments attached to an Invoke, and protocol revenue collected by the application would not automatically be subject to the 50% burn.

The burn applies only to the network fee.

There should therefore be no fundamental accounting conflict at the smart-contract level.

However, an ecosystem-wide audit should take place before activation because some applications may have hard-coded minimum fees such as:

- 100000 WAVELET
- 300000 WAVELET
- 500000 WAVELET

Wallets, bots, and dApp infrastructure should migrate away from hard-coded fee assumptions and use the node's active fee calculation as the source of truth.

13. Sponsored Fee Transactions

Sponsored fees should continue to function.

Under the current model, a user can pay transaction fees in a sponsored asset while the sponsor ultimately covers the corresponding network cost in WAVES.

DFMP should preserve this mechanism.

Current Minimum WAVES Fee -> Effective Multiplier -> Sponsored Asset Conversion

Under x10 or x20, the WAVES-equivalent network fee paid by the sponsor is then divided into 50% burn and 50% generator.

The sponsored application token itself should not be burned.

Asset removed from supply = WAVES

At x1, sponsorship should continue operating under the existing no-burn fee economics.

14. Competitiveness Against Other Networks

The goal should not necessarily be to make Waves the absolute cheapest blockchain in the market.

A healthier objective is:

> **Transaction fees that remain very low for users while still being economically meaningful for the network.**

Using low-fee networks as rough reference points, Polygon transactions can frequently fall around the low fractions-of-a-cent range, while other L2-style environments can also offer transactions in the sub-cent to several-cent range.

Network / Transaction	Approximate cost
Polygon typical low-cost tx	~$0.002
opBNB-style low-cost transfer	~$0.001-$0.005
Waves x20 Transfer	~$0.0043
Waves x20 Exchange	~$0.0130
Waves x20 Invoke	~$0.0216

These are not equivalent workloads. An EVM gas-based transaction cannot be directly compared to Waves transaction-type pricing on a compute-for-compute basis.

The comparison is useful only from the user's perspective of nominal cost. Under that comparison, x20 Waves remains clearly within the low-fee blockchain category at the current WAVES price. A Transfer transaction still costs less than half a cent.

15. Lessons from WEP-5 and Earlier Waves Fee Discussions

Fee reform is not a new subject in the Waves ecosystem.

The 2019 WEP-5 discussion argued that some network fees were no longer aligned with their economic purpose and proposed increasing selected fees partly to improve miner profitability and reduce spam.

Another discussion around a New Fee Calculation Model raised several points that remain relevant today.

1. Miner fee income matters

The economic sustainability of running a node is part of network security.

2. Fee structure should continue reflecting resource usage

This is why the proposed model preserves the relative fee schedule instead of assigning the same fee to every transaction type.

3. Higher fees can reduce transaction demand

This is why the x10/x20 economic model must include transaction-volume sensitivity rather than simply multiplying current fee income by the new multiplier.

Earlier discussion also raised the possibility that miners could have influence over minimum fee levels through voting. DFMP turns that concept into a concrete mechanism based on a governance model Waves already uses today for block rewards.

16. Risks

Transaction Demand Destruction

The most obvious economic risk is a reduction in network activity after fees increase. However, under the x20 + 50/50 model, generator transaction-fee revenue breaks even at approximately 10% of today's fee-generating activity. This creates a large buffer.

Governance Oscillation

x20 -> x10 -> x20

Frequent movement between multiplier levels could create unnecessary complexity for wallets, bots, and application operators. A minimum policy term after each change would reduce this risk. The existing block reward system already provides a useful precedent for slow-moving monetary parameters rather than continuous repricing.

Generator Incentive Conflict

Generators directly benefit from higher fee levels, creating an incentive to vote for larger multipliers. However, this incentive is naturally limited by demand. If transaction activity leaves the network because fees become too high, fee income falls as well. Long-term generator returns also depend on ecosystem activity and the WAVES economy, not simply the nominal fee multiplier.

Protected Registry Abuse

Every project may have an incentive to declare its contract critical infrastructure. This makes a high governance threshold for adding exemptions essential.

Hard-Coded Fees

Wallets, bots, and dApps using hard-coded fees may fail or overpay after activation. An ecosystem audit should therefore precede Mainnet activation.

Sudden WAVES Price Movements

Block voting cannot react instantly to large changes in the WAVES/USD price. This mechanism should therefore not be treated as a real-time USD peg. The multiplier is a medium-term monetary-policy parameter.

17. Proposed Technical WEP Specification

Feature Name

Dynamic Fee Monetary Policy - DFMP

Consensus State

Current Multiplier in {1, 10, 20}

Feature Activation

Mainnet feature approval: 8,000 / 10,000 blocks

Monetary-Policy Voting

Every generated block carries a value such as:

feeMultiplierVote = 1 | 10 | 20

Voting interval: 10,000 blocks

Threshold: 5,001 blocks

If no multiplier reaches the threshold, the current multiplier remains unchanged.

Initial State

At the current WAVES price, initialMultiplier = 20 is the preferred starting point.

A more conservative deployment could activate DFMP at x1 and allow generators to move to x20 during the first fee-policy voting period.

18. Fee Calculation

For normal transactions:

Required Fee = Current Required Fee x Current Multiplier

For protected transactions:

Effective Multiplier = min(Current Multiplier, Protected Cap)

Required Fee = Current Required Fee x Effective Multiplier

19. Burn Calculation

Effective multiplier = 1

$$burn = 0$$

$$generatorPool = fee$$

Effective multiplier > 1

$$burn = fee / 2$$

$$generatorPool = fee - burn$$

The generator pool is then distributed according to the existing Waves-NG ratio:

$$currentGenerator = generatorPool \times 40\%$$

$$nextGenerator = generatorPool - currentGenerator$$

Therefore:

x1: 0% Burn + 40% Current + 60% Next

x10 / x20: 50% Burn + 20% Current + 30% Next

20. API Changes

The node API should make the fee monetary-policy state directly observable.

```
{
  "currentMultiplier": 20,
  "votingInterval": 10000,
  "votingThreshold": 5001,
  "votes": {
    "x1": 600,
    "x10": 2400,
    "x20": 7000
  },
  "burnPercent": 50,
  "generatorPercent": 50
}
```

Block and monetary statistics could also expose:

- totalFee
- burnedFee
- generatorFee
- currentGeneratorFee
- nextGeneratorFee
- cumulativeFeeBurn

This would allow node operators, developers, and analytics platforms such as WavesOnChain to independently monitor the monetary impact of the feature. Transparency is especially important here because the mechanism directly changes both node revenue and WAVES supply.

21. Migration and Testing Requirements

Before activation, at minimum the following invariants should be tested:

Area	Expected behavior
x1	Burn always equals 0
x10	Fee splits 50/50
x20	Fee splits 50/50
Waves-NG	Generator pool keeps 40/60 distribution
Unit0	Protected calls always use x1
Oracle	Configured multiplier cap is respected
Sponsored fee	WAVES side is burned/distributed correctly
Failed Invoke	Existing charging semantics remain intact
Exchange	Matcher network fee calculated correctly
UTX	Transactions below new minimum removed/rejected
calculateFee	Returns active multiplier-aware requirement
rollback	Burn accounting remains deterministic
reorg	Fee carry and burn remain deterministic

Previous WEP-5 discussions already identified the migration issue created when previously valid UTX transactions fall below a newly activated minimum fee. DFMP should explicitly define that behavior rather than leaving it as an implementation detail.

22. Final Recommendation

The central problem with Waves transaction fees today is not that they are expensive.

It is that, at the current WAVES price, they are economically close to insignificant.

The final structure I propose is:

x1 / x10 / x20 Node Voting

x1: 0% Burn + 100% Generator

x10 / x20: 50% Burn + 50% Generator

while preserving Waves-NG:

x10 / x20: 50% Burn + 20% Current + 30% Next

The feature itself should require the normal Waves 80% feature activation threshold. Once active, the multiplier becomes a monetary-policy parameter controlled through 5,001 / 10,000 block voting.

According to the latest 30 completed days in the WavesOnChain fee dataset, Waves currently produces approximately:

238 WAVES/day in transaction fees

At unchanged network activity, x20 would produce approximately:

2,380 WAVES/day burned

2,380 WAVES/day in generator fee income

This corresponds to approximately:

869,000 WAVES/year of activity-linked burn capacity

At the same time, aggregate transaction-fee income for generators rises by approximately 10x.

The model is also not completely dependent on constant transaction activity. If fee-generating activity falls to only 25% of today's level under x20, aggregate generator fee income still remains approximately 2.5x today's level.

This creates a second monetary-policy tool alongside the Waves block reward mechanism.

Block rewards issue WAVES to support network security.

Transaction fee burn removes WAVES when the network is actually used.

Generator nodes participate in governing both systems through block-weighted voting.

Finally, Unit0 should be treated separately from ordinary user activity. Applying x10 or x20 to high-frequency Unit0 consensus traffic is not necessary to achieve the goals of this reform.

Keeping consensus-critical Unit0 transactions under an x1/no-burn regime prevents the upgrade from worsening an existing infrastructure cost problem, while allowing the rest of the Waves network to adopt a stronger fee economy.

The most balanced implementation is therefore:

x1/x10/x20 Community Fee Voting
x1 No Burn
x10/x20 50% Burn
50% Generator Pool
Waves-NG 40/60 Preservation
Unit0 x1/no-burn Protection
Oracle/Infrastructure Caps
5,001/10,000 Monetary-Policy Voting

This design gives Waves the opportunity to achieve three meaningful economic improvements within the same protocol upgrade: stronger node revenue, a sustainable activity-linked WAVES burn mechanism, and protection for high-frequency critical infrastructure from disproportionate fee increases.

References

Waves Documentation - Transaction Fees https://docs.waves.tech/en/blockchain/transaction/transaction-fee

Waves Documentation - Transfer Transaction https://docs.waves.tech/en/blockchain/transaction-type/transfer-transaction

Waves Documentation - Exchange Transaction https://docs.waves.tech/en/blockchain/transaction-type/exchange-transaction

Waves Documentation - Invoke Script Transaction https://docs.waves.tech/en/blockchain/transaction-type/invoke-script-transaction

Waves Documentation - Data Transaction https://docs.waves.tech/en/blockchain/transaction-type/data-transaction

Waves Documentation - Mining Reward / Community-Driven Monetary Policy https://docs.waves.tech/en/blockchain/mining/mining-reward

Waves Documentation - Units / Unit0 Architecture https://docs.waves.tech/en/blockchain/units

Waves Community Forum - WEP-5: Change of Minimum Transaction Fee https://forum.waves.tech/t/wep-5-change-of-minimum-transaction-fee/16294

Waves Community Forum - Proposal: New Fee Calculation Model https://forum.waves.tech/t/proposal-new-fee-calculation-model/13601

WavesOnChain - Network Daily Total Fee https://wavesonchain.com/dashboard/on-chain/waves/daily-total-fee

WavesOnChain API - Daily Total Fee https://beta-api.wavesonchain.com/v1/on-chain/waves/daily-total-fee

Solana Documentation - Transaction Fees https://solana.com/docs/core/fees

Polygon Documentation https://docs.polygon.technology/

Data note: The 30-day fee baseline and the August 26, 2026 WAVES price used in this paper are derived from the WavesOnChain dataset supplied for this analysis. The 30 completed-day window is July 27 through August 25, 2026.