Scaling Definitions

1) Consensus (Old)

Define

T_R = 3000 bytes ; Reference Transaction weight. Note: T_R must be greater than T_2. T_2 equals the weight in bytes of a 2 input and 2 output transaction.

Z_M = 300000 bytes ; Minimum penalty free zone.

M_B = Block weight in bytes.

M_L = The median over the last 100000 blocks of max((min (M_B , $1.7M_L$), Z_M , $M_L/1.7$) ; recursive calculation for M_L with M_L starting at M_L of previous 100001 block (currently = Z_M); Long term median

M_L ; Dynamic penalty free zone.

M_S = the median over the last 100 blocks of max(M_B , M_L) ; Effective short term median.

M_N = min(M_S , $50M_L$) ; Median for Penalty calculation.

R_{Base} = Block Reward.

$0 < M_B \leq 2M_N$; Requirement for valid block.

$B = M_B / M_N - 1$ where $-1 < B \leq 1$

$P_B = R_{Base}B^2$ for $B > 0$; Monero applies a penalty P_B, to increase the block weight by B; $P_B = 0$ for $B \leq 0$

Proposed Scaling Definitions (July 2025 update)

1) Consensus (New)

Define

T_R = 10000 bytes ; Reference Transaction weight. Note: T_R must be greater than T_2. T_2 equals the weight in bytes of a 2 input and 2 output transaction.

Z_M = 1000000 bytes ; Minimum penalty free zone.

M_B = Block weight in bytes.

M_L = The median over the last 100000 blocks of max((min (M_B , $2M_L$), Z_M , $M_L/2$) ; recursive calculation for M_L with M_L starting at M_L of previous 100001 block (currently = Z_M); Long term median

M_L ; Dynamic penalty free zone.

M_S = the median over the last 100 blocks of max(min(M_B , $16M_L$) , M_L) ; Effective short term median.

$M_N = M_S$; Median for Penalty calculation.

R_{Base} = Block Reward.

$0 < M_B \leq 2M_N$; Requirement for valid block.

$B = M_B / M_N - 1$ where $-1 < B \leq 1$

$P_B = R_{Base}B^2$ for $B > 0$; Monero applies a penalty P_B, to increase the block weight by B; $P_B = 0$ for $B \leq 0$

Changes

1) Maximum growth of M_S is reduced from $50M_L$ to $16M_L$
2) Rate of growth and decline of M_L is increased from 1.7x to 2x
3) T_R is increased from 3000 bytes to 10000 bytes
4) Z_M is increased from 300000 bytes to 1000000 bytes

2a) Minimum Fee For Node Relay (Old)

We add a, penalty attracting, transaction T with a size of T_T to a block of weight M_B

Define

$B_T = T_T / M_N$

$P_{BT} = R_{Base}(B+B_T)^2 = R_{base}(B^2 + 2BB_T + B_T^2)$; The new penalty, where $B + B_T > 0$

$P_T = P_{BT} – P_B = R_{Base}(2BB_T + B_T^2)$; Increase in penalty from adding transaction T

$F_T = R_{Base}(2BB_T + B_T^2)$; The additional fee required to overcome the increase in penalty P_T

For the case B = 0 this reduces to $F_T = R_{Base}B_T^2$

$M_F = M_L$; Median for minimum fee calculation

To calculate the minimum fee we consider a transaction of weight T_R at the start of the penalty, B = 0 with $M_N = M_F$ 95% of the fee required to pay the penalty incurred is the minimum fee.

$B_{RL} = T_R / M_F$;

$F_R = R_{Base}B_{RL}^2$; Fee required to pay the penalty incurred

$f_R = R_{Base}B_{RL}/M_F$; Fee required to pay the penalty incurred per byte for a given M_F

$f_I = 0.95f_R$; Minimum fee per byte

2a) Minimum Fee For Node Relay (New)

We add a, penalty attracting, transaction T with a size of T_T to a block of weight M_B

Define

$B_T = T_T / M_N$

$P_{BT} = R_{Base}(B+B_T)^2 = R_{base}(B^2 + 2BB_T + B_T^2)$; The new penalty, where $B + B_T > 0$

$P_T = P_{BT} – P_B = R_{Base}(2BB_T + B_T^2)$; Increase in penalty from adding transaction T

$F_T = R_{Base}(2BB_T + B_T^2)$; The additional fee required to overcome the increase in penalty P_T

For the case B = 0 this reduces to $F_T = R_{Base}B_T^2$

$M_F = M_L$; Median for minimum fee calculation

To calculate the minimum fee we consider a transaction of weight T_R at the start of the penalty, B = 0 with $M_N = M_F$ 95% of the fee required to pay the penalty incurred is the minimum fee.

$B_{RL} = T_R / M_F$;

$F_{RL} = R_{Base}B_{RL}^2$; Fee required to pay the penalty incurred

$f_{RL} = R_{Base}B_{RL}/M_F$; Fee required to pay the penalty incurred per byte for a given M_F

$f_{IL} = 0.95f_R$; Minimum fee per byte

Additional, minimum fee, node relay requirements for large transactions, if such transactions are permitted by consensus

If a transaction has a weight T_T > 10000 bytes
and / or more than 8 inputs
and / or more than 8 outputs

$f_{IN} = 4f_{IL}$

If a transaction has a weight T_T > 20000 bytes
and / or more than 16 inputs
and / or more than 16 outputs

$f_{IM} = 16(M_F/Z_M)f_{IL}$

If a transaction has a weight T_T > 40000 bytes
and / or more than 32 inputs
and / or more than 32 outputs

$f_{IH} = 200(M_F/Z_M)f_{IL}$

2b) Wallet Fees (Old)

For the calculation of wallet fees we assume that the next 10 blocks have no transactions, other than the coinbase transaction, the empty blocks, We then calculate M_{LW} and M_{SW} by following the calculation of M_L and M_S at this future point. We use the previous 99990 blocks and the future 10 empty blocks (100000 blocks) for M_L and the previous 90 blocks and future 10 blocks (100) blocks for M_S.

Define

$M_{BW} = M_B$ for the last 99990 blocks
$M_{BW} = 0$ for the future 10 blocks; A value of 0 bytes can be used for the empty blocks for the purposes of calculating M_{LW}.

M_{LW} = The median over the last 99990 blocks and future 10 blocks (100000 blocks) of max((min (M_{BW} , $2M_L$), Z_M , $M_L/2$) ; The current value of M_L from consensus is used; Effective long term median for wallet fees

M_{LW} ; Penalty free zone for wallet fees

M_{SW} = the median over the last 90 blocks and future 10 blocks (100 blocks) of max(M_{BW} , M_{LW}); Effective short term median for wallet fees

$M_{NW} = \min(M_{SW} , 50M_{LW})$

$M_{FW} = M_{LW}$; Median for wallet fee calculation

$B_{RLW} = T_R / M_{FW}$; Used for the low and normal fees

$B_R = T_R / Z_M$; Used for the medium and high fees

$F_L = R_{Base}B_{RLW}^2$; Low transaction fee for reference transaction

$f_L = R_{Base}B_{RLW}/M_{FW}$; Low transaction fee per byte for a given M_{FW}

$f_N = 4f_L$;Normal transaction fee per byte for a given M_{FW}

$f_M = 16 R_{Base}B_R/M_{FW}$; Medium Transaction fee per byte for a given M_{FW}

$f_P = 2R_{Base}/M_{NW} = f_M M_{FW}/(8B_R M_{NW})$; Maximum Penalty (B =1) Transaction fee per byte for a given M_{NW}

$f_H = 4f_M \max (1, M_{FW}/(32B_R M_{NW}))$; High Transaction fee per byte

2b) Wallet Fees (New)

For the calculation of wallet fees we assume that the next 1000 blocks have no transactions, other than the coinbase transaction, the empty blocks, We then calculate M_{LW} by following the calculation of M_L at this future point. We use the previous 99000 blocks and the future 1000 empty blocks (100000 blocks) for M_L.

Define

$M_{BW} = M_B$ for the last 99000 blocks
$M_{BW} = 0$ for the future 1000 blocks; A value of 0 bytes can be used for the empty blocks for the purposes of calculating M_{LW}.

M_{LW} = The median over the last 99000 blocks and future 1000 blocks (100000 blocks) of max((min (M_{BW} , $2M_L$), Z_M , $M_L/2$) ; The current value of M_L from consensus is used; Effective long term median for wallet fees

M_{LW} ; Penalty free zone for wallet fees

$M_{FW} = M_{LW}$; Median for wallet fee calculation

$B_{RLW} = T_R / M_{FW}$; Used for the low and normal fees

$B_R = T_R / Z_M$; Used for the medium and high fees

$F_L = R_{Base}B_{RLW}^2$; Low transaction fee for reference transaction

$f_L = R_{Base}B_{RLW}/M_{FW}$; Low transaction fee per byte for a given M_{FW}

$f_N = 4f_L$;Normal transaction fee per byte for a given M_{FW}

$f_M = 16 R_{Base}B_R/M_{FW}$; Medium Transaction fee per byte for a given M_{FW}

$f_P = 2R_{Base}/M_{FW}$ Maximum Penalty (B =1) Transaction fee per byte assuming $M_S = M_{FW}$

$f_H = f_P$; High Transaction fee per byte

Changes

1) All fees including the high fee are now based upon M_{LW} with the ratio between fees constant for a given M_{LW}.
2) The grace period is increased to 1000 blocks.

3) Transitional considerations for Reference transaction, T_R, Minimum penalty free zone, Z_M, and Median calculations after the fork.

Define:

Z_{MOld} = 300000 bytes (Z_M before hard fork)

M_{BOld} = Block Weight in bytes (before hard fork)

3) Transitional considerations for Reference transaction, T_R, Minimum penalty free zone, Z_M, and Median calculations after the fork.

Calculation of M_L, M_S, M_{LW} and M_{SW} where blocks from a previous the Monero version are included in a calculation after the fork. M_B is modified as follows:

T_R = 10000 bytes

Z_M = 1000000 bytes

For blocks before the hard fork
$M_B = M_{BOld} (Z_M / Z_{MOld})$

The medians are then calculated normally.

4) Wallet Fee Rounding

Wallet fees, f_N, f_L, f_M, and f_H are rounded up to the desired number of significant digits in the significant

Wallet Fee Rounding Examples

Two significant digits

27810	Rounded to : 28000
37.94	Rounded to : 38
0.5555	Rounded to : 0.56
0.002342	Rounded to : 0.0024

4) Wallet Fee Rounding

Wallet fees, f_N, f_L, f_M, and f_H are rounded to 2 significant digits in the significant

Wallet Fee Rounding Examples

Two significant digits

27810	Rounded to : 28000
37.94	Rounded to : 38
0.5555	Rounded to : 0.56
0.002342	Rounded to : 0.0023

5) Transaction Weights

Transaction weighs are used to account for the different growth rate the output proof verification time with the number of outputs. This is done at consensus level and can lead to double charging if the fee per byte rate is increased because of an increase in the transaction weight.

5) Transaction Weights (Proposed)

Breakdown the transaction weight as follows:

1) Base transaction weight:

Use a standard weights based upon the current size in bytes.

2) Input proofs weight:

Use a standard weights based upon the current FCMP++ proof size in bytes and the number of inputs.

1 input: Current FCMP++ proof size for 1 input

2 inputs: Current FCMP++ proof size for 2 inputs

3 or 4 inputs: Current FCMP++ proof size for 4 inputs

5, 6, 7 or 8 inputs: Current FCMP++ proof size for 8 inputs

More than 8 inputs if applicable: Round up to next power of 2; namely 16 inputs, 32 inputs, etc. and use the current FCMP++ proof size for the corresponding power of 2 number of inputs.

3) Output proofs weight:

Use a standard weights based upon the current proof size in bytes and the number of outputs.

2 outputs: Current proof size for 2 outputs

3 or 4 outputs: Current proof size for 4 outputs

5, 6, 7 or 8 outputs: Current proof size for 8 outputs

More than 8 outputs if applicable: Round up to next power of 2; namely 16 outputs, 32 outputs, etc. and use the current proof size for the corresponding power of 2 number of outputs.

4) TX extra weight:

Use actual size.

The 1), 2) and 3) standard weights do not change if there is a change in the proof size due to significant adoption (this can happen with FCMP++) or a future hard fork. Verification time is then priced at node relay using the multiple node relay fees.