

Hello, Super Validators

3Trade, an application development partner associated with Cumberland, reports that since we completed the implementation of CIP-0002 via on-chain votes on DevNet and TestNet on 30 April 2024, the free traffic available on DevNet is now too small to allow them to complete a transaction to purchase new traffic.

~~This means either that the calculations and simulations performed by Digital Asset in preparation for this change underestimated the differences in fees between DevNet and TestNet, or 3Trade is topping up its traffic balance more often than anticipated.~~

~~Either way, it looks like we should increase the free traffic available on DevNet.~~

After some investigation, Digital Asset has concluded that this is happening to all Validator nodes when they first join the network. Validators are using up their free tier before they are able to complete a minting action or create a traffic balance.

## CIP 0008

### Update and Refinement to CIP-0002

Summary:

This proposal doubles the amount of free traffic (the burst amount) available in a given free traffic window (the burst window), to allow application developers to continue development work on DevNet.

#### Motivation

We want to allow users enough free traffic to allow them to process the transactions that they use to create and update a traffic balance, without allowing so much free traffic that users don't need to create a traffic balance at all.

#### Rationale

The current configuration tilts too far toward limiting free traffic on DevNet, and we need to loosen this constraint.

For background, here's a summary of what CIP-0002 was intended to do, and where we ran into problems:

The key points in [CIP-0001](#), [CIP-0002](#), [CIP-0003 \(revised\)](#) relevant to this proposal are the statements:

- "In order to transfer Coin, the user must spend ~\$1 in Domain Traffic"

- "Calibrate the Free Tier to allow users without any balance to collect rewards and buy domain traffic. Free tier transaction rate is 110% of what's needed to claim rewards and purchase extra traffic in every round."

This meant that the \$/MB of the Synchronizer Fee needed to be scaled so that a transfer of Canton Coin would cost the equivalent of \$1 in terms of traffic in the submitter's traffic balance, and then the round length, free tier, and topup rate needed to be adjusted.

After testing, Digital Asset determined that Canton Coin transfers on TestNet currently use approximately 60kB of the submitter's traffic. This led to a recommendation to set the Synchronizer traffic price (the Synchronizer fee) at \$16.67/MB.

This led to the following recommendation for TestNet and MainNet

Given a traffic price of \$16.67/MB

Given a window of 20 minutes,

Allow the Party 200 kB of free data, and

Automatically top up 200kB of traffic for that Party in those 20 minutes.

So that translated into the following changes for the associated values:

- 60kb per transaction on TestNet
  - Extratraficprice:
    - Before: \$1.00 -> After: \$16.67
- Base rate traffic burst window setting:
  - decentralizedSynchronizer.fees.baseRateTrafficLimits.burstWindow.microseconds:
    - Before: 600,000,000 -> After: 1,200,000,000
  - Calculation:
    - $= 20 * 60 * 1000000$   
(20 minutes in microseconds)
- Minimum topup amount:
  - decentralizedSynchronizer.fees.minTopupAmount:
    - Before: 10,000,000 CC -> After: 200,000 CC
- Burst Amount:
  - decentralizedSynchronizer.fees.baseRateTrafficLimits.burstAmount:
    - Before: 2,000,000 CC -> After: 200,000 CC

Note the scale of these changes:

- Traffic price went up 16.67 times
- The time window for using free traffic went up by 2x
- The default topup amount decreased by 98%
- The free data available during this time period decreased by 90%

TestNet and MainNet must always have the same SV configuration, so a setting that matches our target on TestNet will be valid on MainNet.

Digital Asset noted that this would give slightly different results on DevNet, because there are more SVs on DevNet, which makes transactions somewhat more expensive. But the tests had not been run separately on DevNet, and the difference was expected to be small. This seems to be the root cause of the mismatch we need to resolve in this CIP, though it is also possible that 3Trade is simply topping up so often that they are using up their free tier within each round. The motivation for attempting to use the same settings on DevNet was to avoid having configuration settings from DevNet bleed into TestNet and MainNet unintentionally through operator error.

To resolve this we will need to change one setting on DevNet.

Long term, to avoid errors like this, and to protect against unintended configurations being adopted by the SVs on MainNet, it may be a good idea to restrict DevNet to the same number of SV nodes that are on TestNet and MainNet, plus one or two additional nodes that are preparing to join MainNet at any given time.

## Details

CIP-0008 will increase

`decentralizedSynchronizer.fees.baseRateTrafficLimits.burstAmount`

from

200,000 bytes

to

400,000 bytes

The proposal will not alter the length of time when a given allotment of free traffic can be used, which is represented by

`decentralizedSynchronizer.fees.baseRateTrafficLimits.burstWindow.microseconds`

This change will be implemented via an onchain vote of Super Validator node operators. It involves a change to a single value in a vote proposal.

The vote proposal will be formed by altering the

`decentralizedSynchronizer.fees.baseRateTrafficLimits.burstAmount`

field in the Action in the SV Governance window [Add DSO App Configuration Schedule](#), as shown here:

decentralizedSynchronizer.fees.baseRateTrafficLimits.burstAmount	200000
decentralizedSynchronizer.fees.baseRateTrafficLimits.burstWindow.microseconds	1200000000

Operators who open the vote proposal will see a data structure with the “burstAmount” value, shown as 200000 in the screenshot below, changed to 400000.

```
"fees": {  
  "baseRateTrafficLimits": {  
    "burstAmount": "200000",  
    "burstWindow": {  
      "microseconds": "1200000000"
```