File Storage

DashDrive uses a system of file storage based on the concept of distance from the common root node of the first few characters of all coinbase addresses. The premise of this storage mechanism is that all addresses are in the same logical address space. As as example we will compute the first five addresses in the dash blockchain, then we will show how the network can calculate where the storage information is at and how to retrieve and execute updates based on those positions.

Example 1:

XtuVUju4Baaj7YXShQu4QbLLR7X2aw9Gc8

XbmPwcZCdaRMa4m4JedspBPkUqY1FKRhsX

XgfLsphG9KDGmyBEwHmBjMRYa8Hr7Yode5

XxYTj1hgGMx6X6PgFmTsq4zcMkhrSMczTJ

Xo9UzA2B7yyEhsrQZyFPGCTuFbNGKgHkPU

Hopping through the network

5 XtuVU -> 1st hop

10 ju4Ba -> 2nd hop

15 aj7YX -> 3rd hop

20 ShQu4 -> 4th hop

25 QbLLR -> 5

30 7X2aw -> 6

First operation is to calculate the distance of “XtuVU” from “XAAAA”, the first logical possible address in the space. This calculates two numbers, then we substract one from the other to calculate the offset of storage position. We then declare the storage positions on the network and which numbers they start and stop at.

We use a python module for base conversion of our strings (base58) to decimal.

$ADDR1 XtuVU 247773176

$BASIS XAAAA 237646416

$XFF = 10126760

To calculate which dashdrive sector we store in, we calculate an offset modulus value, then get the offset and that is our storage vector.

$TIER\_N\_VECTOR\_MOD = $B % 7