ELECTRIC LIGHT NAME SERVICE (ELNS)

A PREPRINT

Author's SHA1 hash: 96dda4ffa881d25c94163d25569f4164e626901f

10-26-2020

ABSTRACT

Electric Light Name Service (ELNS) is a proposed utility to resolve IPNS hashs by mapping them to human readable words. The mapping will function similar to the way DNS maps IP addresses to domain names. However, ELNS will have one major difference, ELNS-names are not chosen, requested, or purchased by the participant. Instead, nodes participating in the network will be awarded a single randomly chosen ELNS-name. Nodes will then be required to participate on the network to maintain their name. The core idea behind the system is to reward participation without financial compensation or require participants to pay into the system to purchase names. We also want to keep the barrier to entry as low as possible to allow an even playing field. This system should allow anyone with a computer, able to run IPFS, to participate and be awarded a name.

1 Introduction

Inter-Planetary File System (IPFS) is growing in adoption but it's use of content identifiers (CID) can be hard for humans to remember. These CID's are, currently, 46 characters long[1] and IPFS does not have a way to map them into human friendly names like Domain Names map to IP's. A few projects have been created to address this, however, each of them have their own set of issues and limitations[2].

The CID's also have an issue for websites with mutable content because they are based on the contents cryptographic hash. For example, a blog's hash would change with every new post. To address this issue, IPFS has created the Inter-Planetary Name System (IPNS)[3]. From their website: "A name in IPNS is the hash of a public key. It is associated with a record containing information about the hash it links to that is signed by the corresponding private key. New records can be signed and published at any time."[4]

The IPNS names are able to function similar to the way a domain name does. However, they are still cryptographic hashes and thus hard for humans to remember and use. This project proposes to create a new way to map human readable names to IPNS hashes. Similar to the way DNS functions on the Internet. However, unlike DNS, we wish to propose a new way of naming that will reward participation and still allow IPFS to have easy to use, and memorable, names for published IPNS hashes.

The proposed system will also not inhibit participants from extending it to existing naming systems (such as DNS). With the current DNS implementation, mapping Domain Names to IPNS names can be done using TXT records, thus allowing current domain name owners to keep their existing Domain Names. With ELNS, this same method can be used. In future, it might be possible to achieve the same results using something like CNAME records.

[1] *Content Identifiers*. IPFS. URL: `https://docs.ipfs.io/guides/concepts/cid/`.

[2] *Ten terrible attempts to make the Inter Planetary File System human-friendly*. 2017. URL: `https://hackernoon.com/ten-terrible-attempts-to-make-the-inter-planetary-file-system-human-friendly-e4e95df0c6fa` (visited on 09/26/2017).

[3] *Inter-Planetary Name System (IPNS)*. IPFS. URL: `https://docs.ipfs.io/guides/concepts/ipns/`.

[4] *Inter-Planetary Name System (IPNS)*. IPFS. URL: `https://docs.ipfs.io/guides/concepts/ipns/`.

2 Names

Names will not have the ability to be chosen or purchased. They will only be awarded after a node completes and submits a Name Award Request as outlined in Section 3. Once a node's request is approved by the network, a single ELNS name is awarded per node. The ELNS name is then mapped to a nodes published IPFS ID, the transaction is stored on a blockchain and a distributed hash table is updated with the IPNS hash that node publishes. After a node is awarded a name, it will be permanent and not transferable. Hopefully, this will help to deter name mining or name hoarding. A node is allowed to change their IPNS hash but any new IPNS hashes must belong to the same nodes IPFS ID. Nodes are also required to keep participating on the system otherwise their name will be revoked as outlined in Section 4.

We chose this solution for a few reasons.

- As in life, you don't get to choose your name. That is something given to you at birth. Most everyone born has this same experience. This system is designed with that mind.
- Existing DNS names are valued mainly based on their vanity, something people with more money can use to their advantage. Something also used by actors to hoard names and then force others to purchase at a high cost. These two reasons add a barrier to entry for some with the existing DNS system.
- Another issue with DNS is around censorship and anonymity. The current DNS system requires people to purchase their domain names from a Registrar. The Registrars thus have access to the personal details of who owns the domain. This can be used to censor or de-platform people by not allowing them to transact. Removing the financial aspects from ELNS will help ensure that a participant will never need to disclose their personal information. All that is required is a minimum amount of participation.

2.1 Name Format

An ELNS-name is derived from a combination of three randomly chosen words. These words are chosen from a select pool of words for a given language. The words are selected based on various criteria such as: length, commonality, and ease of spelling. This collection will then be pruned to remove various other words like: homophones, offensive, or copyrighted words. These words are then assembled into a Language Dictionary (LD) described in 2.2.

ELNS names use common dot notation. Where a '.' is used to separate each part of the name. The name begins with the i18n code for that language[5], followed by the given words, and ends in "elns".

ELNS name format:

$$< i18ncode > . < first - word > . < second - word > . < third - word > . < elns >$$

For example: `"en.sprinkle.tree.none.elns"` or `"en.aspire.door.meatloaf.elns"`.

2.2 Language Dictionaries

After the collection of words has been agreed upon for a given language, a file will be created in IPFS to store these words. They will be ordered alphabetically, or in the same manner one would order words in a dictionary for that language. Each line in this file will contain a single word. These files will be referred to as Language Dictionaries (LD) and will be named in the following format:

$$< i18ncode > .ld$$

For example: "EN.ld" for English names. Each node participating in ELNS, for that language, will be required to pin this file.

3 Name Award Request (NAR)

Nodes will be granted their name after they have completed a simple puzzle technically outlined in section 3.1 and submit a Name Award Request (NAR). Unlike crypto currencies, the problem being solved is not intended to be

[5] *i18n Internationalization*. i18n. URL: `https://www.w3.org/International/`.

computationally intensive. Instead, the puzzle is intended to ensure the names are generated from random numbers. It is also designed to be easily verifiable by any node on the network.

This puzzle will require the node to produce three random Nonce's. These Nonce's will then be used to select the three words that will compose the nodes name. Only one name can be awarded to an IPFS ID. This is a strict one-to-one mapping and no IPFS ID can have more than one name. A blockchain will be used to store and track the name awards as outlines in Section 5

3.1 Name Award Problem

The algorithm is as follows:

1. The target will be a SHA1 sum that has N leading zeros
2. The first input will be the latest Merkle root from the blockchain
3. Append a nonce to the input to produce a string that has an SHA1 sum less than the target. This will produce the first nonce
4. Use this hash as the input for the next round to find the 2nd nonce
5. Use the hash from the 2nd round as input to find the 3rd nonce
6. Use these nonce's to calculate: w_i = ($nonce_i$ mod (LINES-IN-LD)), $i \in \{1, 2, 3\}$

Name Example:

NOTE: ***this example is for illustration and the hashes may not be correct***

First Round

merkel-root = "00000030af80430252cb3c6937879fe0ddd5c3c9"

$nonce_1$ = "376862b4703d8cc2"

SHA1("00000030af80430252cb3c6937879fe0ddd5c3c9376862b4703d8cc2") $\Longrightarrow hash_1$

$hash_1$ = "00000016b9595c1b80e4186301267a0aa5404710"

Second Round

$hash_1$ = "00000016b9595c1b80e4186301267a0aa5404710"

$nonce$ = "1b9c9e86b71a740f"

SHA1("00000016b9595c1b80e4186301267a0aa54047101b9c9e86b71a740f") $\Longrightarrow hash_2$

$hash_2$ = "00000001d23a08f7a1a59a3946e56705f0741e0a"

Third Round

$hash_2$ = "00000001d23a08f7a1a59a3946e56705f0741e0a"

$nonce$ = "1b9c9e86b71a740f"

SHA1("00000001d23a08f7a1a59a3946e56705f0741e0a1b9c9e86b71a740f") $\Longrightarrow hash_3$

$hash_3$ = "0000000edd3f08f7a1a59a3946e56705f0741e0a"

Name Composition

$nonce_1$ mod (LINES-IN-LD) = 4157, The word in the LD file at line 4157 is: **beetle**

$nonce_2$ mod (LINES-IN-LD) = 2348, The word in the LD file at line 2348 is: **argyle**

$nonce_3$ mod (LINES-IN-LD) = 32968,The word in the LD file at line 32968 is: **resay**

The final name would be: **en.beetle.argyle.resay.elns**

The above could produce name collisions or partial name overlaps. If this occurs, the first host to get their block added to the chain will receive their name. The other hosts will need to try again.

3.2 Lookup Data Base

To help prevent name collisions and allow nodes to verify the name is not already taken, Cuckoo filters will be used[6]. The filter will be updated after every name award round is held. New names will be added and reclaimed names will be removed. This filter will be expected to be pinned by all participating nodes.

4 Name Reclamation Request

Hosts are required to re-validate their name and maintain their entry in the Distributed Hash Table (DHT) as outlined in section 6. If a node fails to keep their records up to date, the name may be reclaimed by the network after 72 hours. Any participating node can submit a Name Reclamation Request (NRR). The network will then validate the request and reclaim the name and return it to the free pool. This is to prevent mining or hoarding names that will never be used. If a node has it's name reclaimed, it can submit a request for a new one.

4.1 Name Reclamation Request (NRR)

A Name Reclamation Request will consist of the following:

- ELNS Name of offending node
- Type: (expired or malicious block)
- Details (expired timestamp or malicious block ID)
- Current timestamp
- ELNS Name of submitting node
- Hash signature of submitting node

The reclamation submissions will take priority over new name awards.

5 ELNS Blockchain

The ELNS blockchain will function as the ultimate record of all names and their IPNS ID mappings. It will be modeled on Bitcoin and Ethereum blockchains but instead of tracking credits and debits of coins, it will track credits and debits of names. We won't go into much detail in this paper on how the blockchain will be implemented. The blockchain technology is now widely known. Instead, we will give a high level overview of how the blockchain will function in general. The blochain will have the following transactions:

- Name Award (credit from pool)
- Name Reclamation (debit to pool)

The pool size will be determined by the size of the LD. If for example, their are 30,000 words in the LD, the pool size will be: $30,000^3$

5.1 New Blocks

New blocks are created by validating NRR and NAR submissions. These are akin to transactions on a blockchain. A dynamic limit on the number of submissions per block will be set to prevent overloading hosts and keep computation at an acceptable amount.

5.2 Blockchain Consensus

Consensus on the blockchain will be done using Proof Of Stake (PoS). Each node will stake their name. This will place every node on an equal footing. The first node to have their block validated will win the round.

Note: If a node submits a malicious block, their name will be forfeit and reclaimed.

[6] Michael Mitzenmacher. *Bloom Filters, Cuckoo Hashing, Cuckoo Filters, Adaptive Cuckoo Filters, and Learned Bloom Filters.* URL: `https://smartech.gatech.edu/handle/1853/60577`.

5.3 Blockchain Participation

To create incentive for nodes to participate on the blockchain, nodes will be required to submit or validate new blocks to the chain. Nodes that successfully wins a new block round will be rewarded with a 72 hour reprieve from requiring to re-validate their name. Nodes who attest new blocks will be rewarded with 12 hours reprieve.

6 Name Resolution and Distributed Hash Table

To help with rapid name resolution, the system will use a Distribute Hash Table (DHT). Each node will be expected to participate in responding and resolving names. The DHT will contain Name Validation Records (NVR) . The NVR will contain:

- ELNS name (key)
- IPFS ID
- Unix timestamp
- Block ID of the attested/created block
- Signed hash of nodes IPFS ID and Block ID using the nodes IPFS public key

6.1 Name Resolution

Each node will be required to participate on the DHT and the burden of participation will be kept to a minimal. Nodes will be expected to:

- Accept and respond to name resolution requests
- Add new names and validate them against the blockchain
- Respond to NVR updates
- Remove reclaimed names
- Act as a local cache for resolved names

Example Name Resolution Request

1. Look up the name in the DHT
2. Inspect the resolved record and ensure it's valid (within the last 12 hours)
3. If the name is valid, return the name along with it's most recent NVR so the end host can also verify the name as outlined in Section 6
4. If the name is invalid, return an error and submit a NRR request outlined in Section 4

7 Scalability

As of today it is estimated the the current total registered domain names is somewhere around 330,000,000[7]. The ELNS naming scheme could easily accommodate into the Trillions of names with just 3 words. We base this estimate on using only a portion of the available words in the English language. As of 1993, the Oxford English Dictionary included around 430,000 words[8] Our estimation is that we will make use of less than 10% of these words. Somewhere in the area of 30,000 in total. This would yield a total of $30,000^3$ total combinations or 27,000,000,000,000 possible names.

NOTE: If a name mapping is removed, or de-awarded, it will return to the pool and could potentially be re-used.

[7] *Verision Domain Name Industry Brief.* 2017. URL: https://blog.verisign.com/domain-names/verisign-domain-name-industry-brief-internet-grows-to-330-6-million-domain-names-in-q1-2017/ (visited on 07/18/2017).

[8] *How many words are there in English?* Merriam-Webster. URL: https://www.merriam-webster.com/help/faq-how-many-english-words.

References

[1] *Content Identifiers*. IPFS. URL: https://docs.ipfs.io/guides/concepts/cid/.

[2] *How many words are there in English?* Merriam-Webster. URL: https://www.merriam-webster.com/help/faq-how-many-english-words.

[3] *i18n Internationalization*. i18n. URL: https://www.w3.org/International/.

[4] *Inter-Planetary Name System (IPNS)*. IPFS. URL: https://docs.ipfs.io/guides/concepts/ipns/.

[5] Michael Mitzenmacher. *Bloom Filters, Cuckoo Hashing, Cuckoo Filters, Adaptive Cuckoo Filters, and Learned Bloom Filters*. URL: https://smartech.gatech.edu/handle/1853/60577.

[6] *Ten terrible attempts to make the Inter Planetary File System human-friendly*. 2017. URL: https://hackernoon.com/ten-terrible-attempts-to-make-the-inter-planetary-file-system-human-friendly-e4e95df0c6fa (visited on 09/26/2017).

[7] *Verision Domain Name Industry Brief*. 2017. URL: https://blog.verisign.com/domain-names/verisign-domain-name-industry-brief-internet-grows-to-330-6-million-domain-names-in-q1-2017/ (visited on 07/18/2017).