A First Look at the Name Resolution Latency on Handshake

Katsuki Isobe Daishi Kondo Hideki Tode

Osaka Metropolitan University

Motivation

- DNS faces various issues, including security and privacy, owing to its traditional architecture [4]
- Blockchain-based DNS has recently emerged to solve these DNS issues

It is necessary to evaluate the performance of the Blockchain-based DNS to assess its suitability as an alternative to the current DNS

Handshake Overview

Handshake [1], one of the blockchain-based DNS, aims to resolve the centralization of the TLD by ICANN by replacing the current DNS root servers with blockchain nodes

- Two types of nodes
  - *Full node* (*full client*: *hsd*): node storing all the blockchain data
    - *Full client* user can resolve Handshake names locally
  - *Light node* (*light client*: *hnsd*): node working as a simplified payment verification (SPV) node
    - *Light client* user can resolve Handshake names by connecting to the *full nodes* and querying the proof of the name to them (Figure 1)
- Two types of P2P connection
  - *Brontide*: encrypted P2P connection (essentially used by *seed nodes*)
  - *Clearnet*: unencrypted P2P connection
- Other notable terms
  - *Seed node*: full node run by Handshake community
    - Every node connects to *seed nodes* when started for the first time
  - *Pool size*: max. # of simultaneous connections

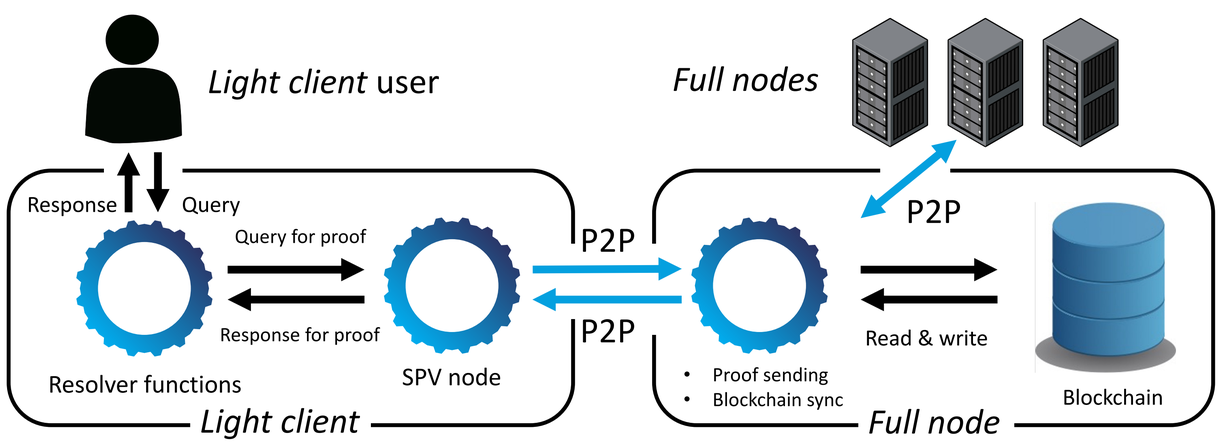

Figure 1. *Light client* behavior for name resolution on Handshake

Experiments and Results

- Setup for latency measurement of Handshake
  - Locations of *light client* running in AWS EC2 instance: Australia (AU), Bahrain (BH), Brazil (BR), Germany (DE), Japan (JP), United States (US (east), US (west)), and South Africa (ZA)
  - TLD dataset (randomly selecting 100 TLDs): generated from volunteer's website [2]
  - # of queries: 100 queries in *pool size* varying from the default value of 8–80 in 8 increments
- Setup for latency measurement of the current root servers
  - Locations of DNS client running in AWS EC2 instance: same as the setup for latency measurement of Handshake
  - TLD dataset (unique top-100 TLDs): generated from Tranco dataset [3]
  - # of queries: 100 × 13 (A∼M root letters) queries
- Results
  - The average on Handshake was approximately 6 times as large as that on current DNS root servers (Figure 2) because the *full nodes* returning responses were not geographically distributed
    - 19, 7, 4, and 4 nodes were in the U.S., China, Germany, and other countries, respectively
    - 8 nodes were *seed nodes* that used *brontide* and they essentially kept connected with our nodes
  - *Brontide* achieved larger latency than *clearnet*
    - The communication ratio with the non-*seed nodes* to that with the *seed nodes* increased as the *pool size* increased (Figure 3)
    - The latency with *brontide* was more than 1.6 times as high as that with *clearnet*, indicating that the encrypted connection might cause overhead

A practical latency drawback exists in Handshake against the current DNS root servers at the current early stage

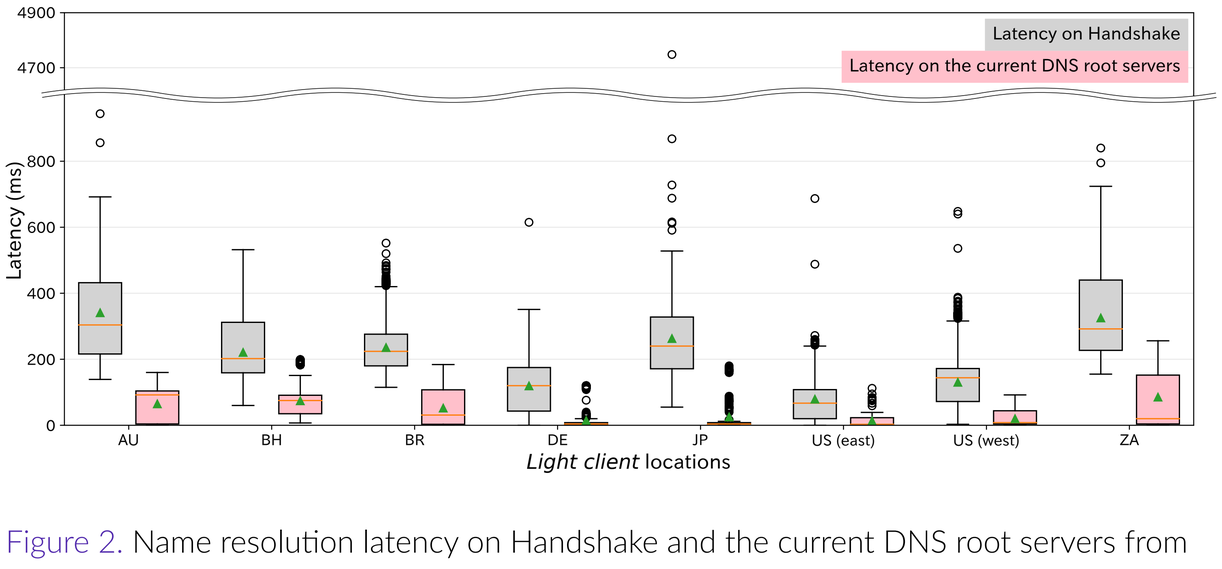

Figure 2. Name resolution latency on Handshake and the current DNS root servers from different locations (the green triangle and orange lines indicate the average and median values, respectively)

Future Works

- A detailed analysis of the cause of the high latency (e.g., the *brontide* overhead)
- A proposal to reduce the latency on Handshake to the same level as the current DNS root servers

References

[1] Handshake. `https://handshake.org`.

[2] Nameboard. Available at `http://adboard.durendil.hns.to` (accessed January 30, 2022).

[3] Tranco list. Available at `https://tranco-list.eu/list/2369/1000000` (accessed January 14, 2022).

[4] G. Schmid. Thirty Years of DNS Insecurity: Current Issues and Perspectives. *IEEE Communications Surveys & Tutorials*, 23(4):2429–2459, 2021.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP21K17741

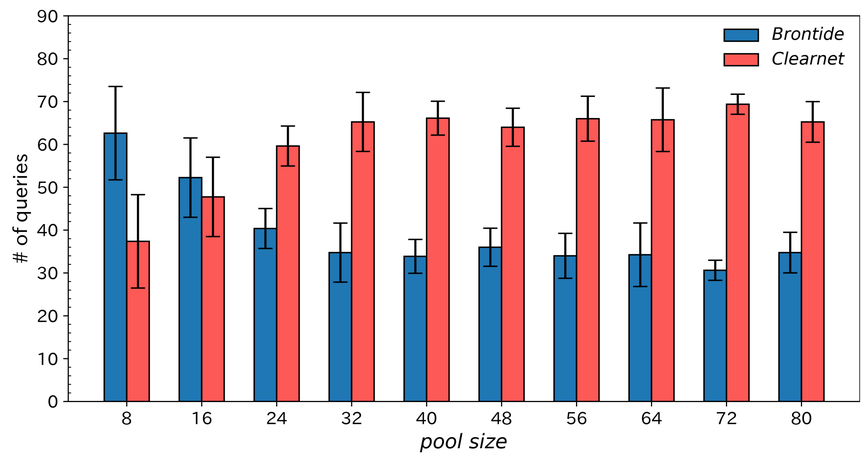

Figure 3. Comparison of the number of queries sent as the proof requests between *brontide* and *cleanet* per *pool size* (error bar: standard deviation)