DIVA.EXCHANGE

DNS for I2P: a Distributed Network without Central Authority

Creating a DNS for an Overlay Network without a Central Authority

Author: Konrad Bächler
Twitter: @DigitalValueX, Web: https://diva.exchange

Agenda

- Concept of the I2P network (7')
- Motivation: Why DNS for I2P on DIVAchain (3')
- Concepts and Context (5')
- Byzantine Fault Tolerance as consensus (5')
- The Good & the Bad since FOSDEM '23 (5')
- Discussion/Feedback (5')

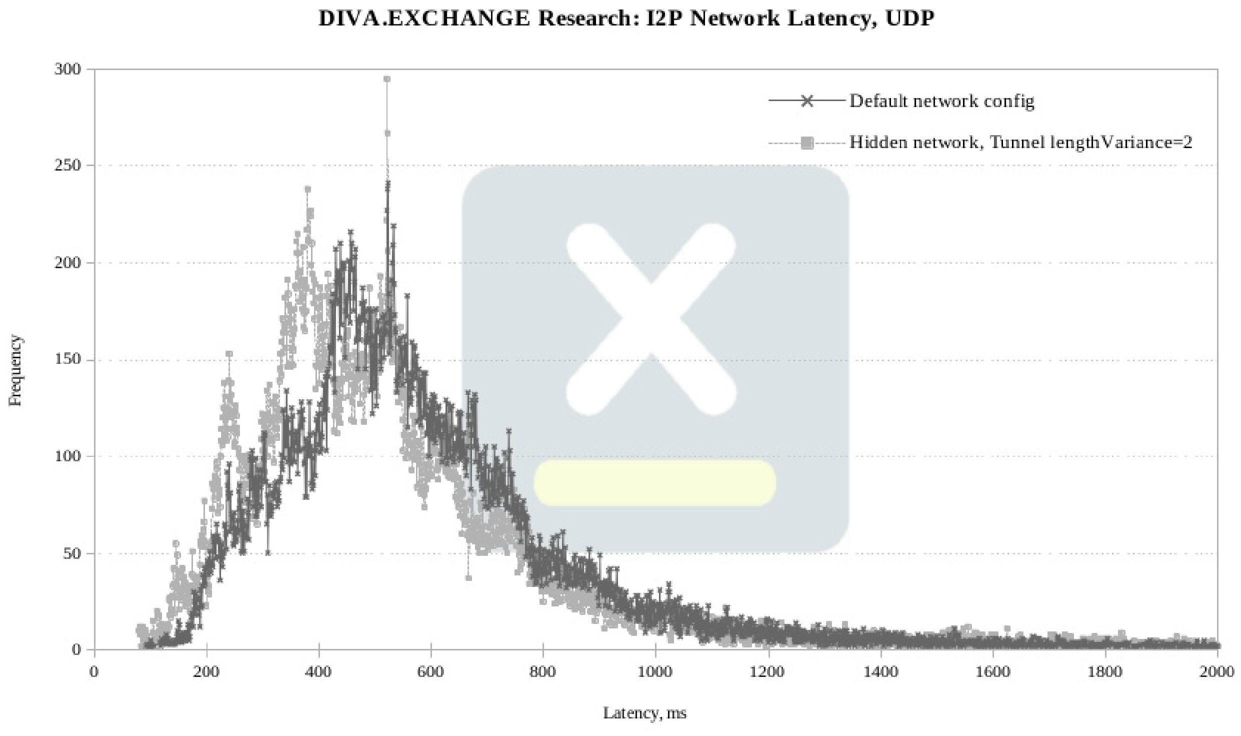
About diva.exchange

- Non profit association, open to everyone
- A loose bunch of Devs & Researchers - spread all over the world
- «DIVA - Free Banking Technology for Everyone» means: handle all kind of Digital Values under your own control and responsibility and apply your very own philosophy of privacy without being nudged by others
- No centralized business model (pointless); no token/coin.

Hello I2P Network

- A few basic facts (some are simplified - educational reasons):
  - I2P is an overlay network (misleading name «darknet» is just used by dubious media desperately in need for clicks)
  - It's a peer-to-peer network where every node in the network acts as a router
  - I2P itself has no storage capabilities – it is a transport layer
  - Messages travelling through the network are multiple times encrypted (like a garlic: it has multiple layers) – call it «Confidentiality feature»
  - Messages hop over several routers within the network to their final destination (using «tunnels») – call it «Anonymity feature»
- In a nutshell: I2P = confidential & anonymous message transport

Latency: I2P network

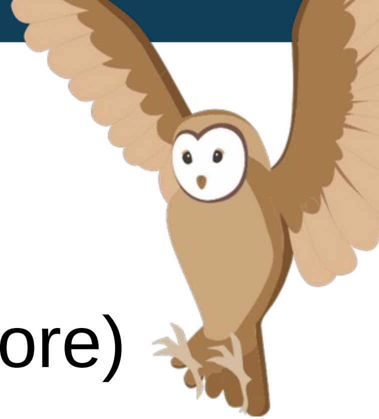


Raw datagrams, 82105 samples, evenly distributed over 6 hours +++ July 2023 +++ created using https://github.com/diva-exchange/i2p-sam

Concept: I2P-b32 Adresses

- Remember: the I2P network provides anonymity
- Example:
  auoqibfnyujhcht4v3nzahpqztwlyomesfywltuls5bqqi3nd3ka = diva.i2p
- Destinations within the network («peers») are identified by public keys
- A Public Key of a destination can be transformed to a b32 address. I2P-b32 address = the base32 encoded sha26-hash of the public key
- There exists no algorithm to resolve a name (like «diva») to it's b32 address – it's only a local lookup in a key/value store

Concept (2)



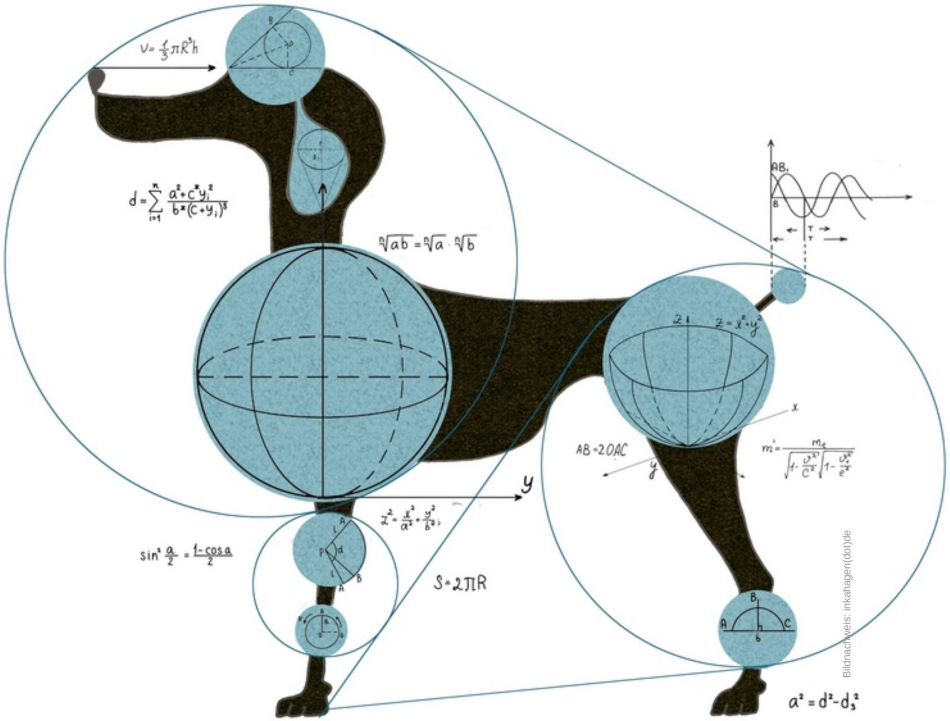
- I2P has only a local addressbook (a key/value store)
- Users can build their own addressbook or download it from a «trusted» source (which is a joke by itself in an anonymous and trustless network)
- DIVA studies a very trivial use case of DNS: mapping destinations to shorter and simpler names

Motivation: Why? (1)

- I2P shall be a fully distributed network – without central authority
- Today, the I2P network has some «jump services and name registries» which act as kind-of-DNS. This is obviously a central component/concept and hence not compatible with the idea of true distribution (not saying that there are bad actors, it's just not compatible).
- True distribution = distributed ledger = blockchain-like system

Why? (2)

- Fully anonymous
- Immutable
- Barrier-free
- Trustless



Bildnachweis: inkahagen(dot)de

Trust me:

Concept of «Trust»

- Example: believing in «Flat Earth» means «trusting a specific set of made-up root data leads to misleading conclusions»
- Typically: Larger systems (technology- or human-driven) building on trust, grow to a complex, strongly regulated and highly intransparent colossus over time. Such systems are wrong by design, mostly corrupted and the opposite of elegant.
- Solution: (a) base systems and decisions on math, keep them lean/small, uncoupled and transparent and (b) add a cost function to prevent abuse

Context (1) - History

- A range of approaches to distribute DNS exist – like an older/failed project: «[…] a **currency native to the blockchain is necessary** to create a cost function […]»
- Fact: DIVAchain **does not implement** concepts like «currency-based cost functions» or «trusted (validator) nodes». Side-note/remember: a non-fungible «currency» isn't a currency.
- Naive DHT-like concepts are failing on «immutability» and «integrity» (same problems as the «trust» concept)

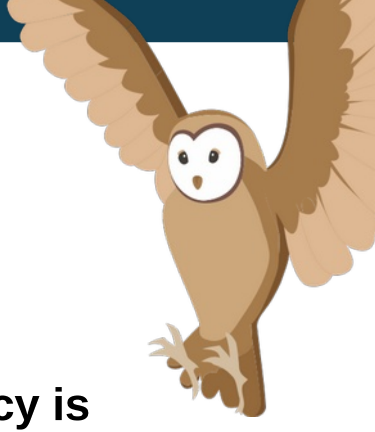
Context (2): CAP theorem

- CAP theorem: Consistency, Availability, Partition tolerance
  - DIVAchain chooses «Availability and Partition tolerance» (aka «eventual consistency»)
- Why? Consistency is not important in this context. Why? Because it's «cheap» to query the network to detect partitioning.

Context (3) - Fallacies

- In distributed computing, there are well-known fallacies. A few:
  - Zero network latency, unlimited bandwidth
  - Homogenous and secure network
- Fact: poorly developed systems will fail by using I2P as a transport layer
- DIVAchain works as expected using I2P as a transport layer

Context (4) - DIVAchain



- Properties
  - Blockchain-like system («CAP theorem»: within the I2P network, **consistency is too expensive**, see also «Fallacies of Distributed Systems»)
  - supports «immutability» and «integrity»
  - byzantine fault tolerant (BFT) approach (no proof-of-work, no validators)
- Cost functions
  - Cost of writing to local: almost zero
  - Cost of replication (=requiring a vote on local data from a remote peer): to be finally specified and implemented (probably: a function of availability and/or co-operation)

The Bad... (1)

- Blockchain implementation: Democratic Byzantine Fault Tolerance (BFT, see FOSDEM 23) **failed**. Reason: I2P reliability (see fallacies) is not sufficient for BFT – example, time to consensus: 32 nodes, >2 minutes.

...and the Good (2)

- The «eventual consistency» approach of DIVAchain seems to finally deliver reasonable results and it becomes applicable for diva.exchange use cases
- Feedback on the experimental DNS project from academia and students was positive and highly motivating
- I2P research (de-anonymization studies) received more attention both from academia and the I2P community
- DIVAchain and I2P test network available

Summary and Take Outs

- DNS for I2P on a modified BFT-blockchain-like system is a reasonable solution approach
- The core challenges, as known today:
  - Implementing cost functions
  - Implementing «decisions», implementing global state
  - It's a very naive (and maybe silly) «first-come-first-served» approach
- Partizipation in the project is much welcome - just fork it and get involved please…

Sources

- Docker compose, source code: https://github.com/diva-exchange/diva-dockerized
- DIVAchain, source code: https://github.com/diva-exchange/divachain
- I2P SAM library, source code: https://github.com/diva-exchange/i2p-sam
- Container / Docker images, including documentation: https://hub.docker.com/u/divax
- I2P Research (like scalability tests, de-anonymization approaches): https://github.com/diva-exchange/academia

DIVA.EXCHANGE

Discussion / Links

Web: https://diva.exchange/

Twitter: @DigitalValueX

Mastodon: @social@social.diva.exchange

Telegram Group: https://t.me/diva_exchange_chat_de

Source Code (AGPL3 or better; Apache 2.0) & Research/Academia: https://github.com/diva-exchange

Docker Images (I2P and DIVA stuff): https://hub.docker.com/u/divax

I2P & Docs: https://geti2p.net

Git repos to fork...