Workgroup: Network Working Group
Internet-Draft:
draft-trossen-rtgwg-impact-of-dlts-02
Published: 30 August 2022
Intended Status: Informational
Expires: 3 March 2023

Authors: D. Trossen	D. Guzman
Huawei Technologies	Huawei Technologies
M. McBride	X. Fan
FutureWei	IoTeX

Impact of DLTs on Provider Networks

Abstract

This document discusses the impact of distributed ledger technologies being realized over IP-based provider networks. The focus here lies on the impact that the DLT communication patterns have on efficiency of resource usage in the underlying networks. We provide initial insights into experimental results to quantify this impact in terms of inefficient and wasted communication, aligned along challenges that the DLT realization over IP networks faces.

This document intends to outline this impact but also opportunities for network innovations to improve on the identified impact as well as the overall service quality. While this document does not promote specific solutions that capture those opportunities, it invites the wider community working on DLT and network solutions alike to contribute to the insights in this document to aid future research and development into possible solution concepts and technologies.

The findings presented here have first been reported within the similarly titled whitepaper released by the Industry IoT Consortium (IIC) [IIC_whitepaper].

Status of This Memo

This Internet-Draft will expire on 3 March 2023.

Copyright Notice

Table of Contents

1. Introduction

The current routing system was initially designed for a single purpose, namely reachability between end nodes. This capability is utilized in many higher layer technologies in the form of overlays. Distributed Ledger Technologies (DLT) are one such form of overlay with the aim to facilitate communication patterns that allow a distributed consensus among distributed, and generally unknown, participants in the DLT overlay.

The realization of a DLT overlay follows that of other well-known examples for distributed computing tasks, such as Torrents, distributed file storage, among others. That is, DLTs form their own overlay through contributing 'peers' that partake in the DLT. For this, reachability information (in the form of IP addresses) of other DLT peers is centrally maintained (in so-called 'bootstrap nodes') to establish peer-specific pools of peers, within which each peer in turn communicates for the specific purpose of the DLT. DLTs secure the transactions using transport-level methods. As an overlay, DLTs are little concerned with the underlying network(s) itself, simply utilizing the provided IP reachability service for their purpose.

Continuing on the insights first reported in [IIC_whitepaper], this document sheds light onto the realization of specific DLT overlay mechanisms from the perspective of the resulting impact on the utilized provider networks in the form of the actual communication taking place.

For this, we outline the communication patterns upon which certain forms of DLTs rely (Section 4.2) in order to implement the key DLT concepts (Section 3). Based on our insights of those communication patterns, we then identify a number of key challenges (Section 5) through initial experimental results (Section 6) within an example DLT platform (here, Ethereum [REF]).

Here, we explicitly recognize that those insights are highly dependent on the specific DLT mechanisms under investigation and are therefore not generally transferable to other DLT platforms and their realization. For instance, DLT platforms relying on proof-of-work for transaction verification tend to differ in their communication from those relying on proof-of-stake. However, this document does attempt to develop a wider methodology over time that may allow for quantifying the impact on underlying networks across those different types of DLTs.

While the quantification of DLT impact serves as an interesting benchmark into the possible costs for operating DLTs, the identified challenges give also rise to possible opportunities for network-level innovations to improve on the situation observed in our experiments, thereby reducing the identified impact on provider network. Section 7 outlines a possible realization of those opportunities through a constraint-based selection of communication relations, utilizing semantic information beyond IP reachability.

With this in mind, we position an improved DLT performance as a possible applicability for semantic routing, introduced in more detail in [I-D.farrel-irtf-introduction-to-semantic-routing], while also soliciting other possible realizations of an improved DLT performance through network-level innovations. Moreover, we draw connections with ongoing IETF/IRTF efforts (Section 8), where our insights may provide useful input.

Note: This document does neither discuss the particular rationale for selecting DLTs in order to realize the intended application purpose nor the specific DLT mechanisms eventually used. It therefore does not pass comment on the advisability or practicality of using DLTs and their solutions, nor does it define any specific technical solutions for reducing the observed provider impact.

2. Terminology

The following terminology is used throughout the remainder of this draft:

Smart contract : distributed state machine over which transactions will be executed and logged.

Transaction : cryptographically signed (set of) instruction(s) against a smart contract.

Ledger : information on transactions

Block : set of verified ledger information

Blockchain : concatenated blocks with longest chain of blocks representing the current consensus of ledger information.

Peer : participant in the DLT, with a possible narrower role of client or miner.

Client : a DLT peer issuing transactions towards a set of miners.

Miner : a DLT peer receiving transactions from miners and performing suitable block operations and exchanges to maintain DLT information.

3. Main DLT Concepts

There has been ample work, such as [DLT_intro] [DLT_intro2], among others, including in other SDOs such as the IEEE but also within the IRTF/IETF [DINRGref], on defining main DLT concepts; we refer the reader to those references for more details. We focus our brief introduction here on those concepts most important to understand from a communication perspective.

The core abstraction used in a DLT is that of a 'transaction', i.e., a cryptographically signed (set of) instruction(s) to modify a state machine, which in turn represents the distributed consensus the DLT is trying to maintain. These transactions are executed within the higher-level concept of a 'smart contract', which implements the specific DLT application, such as for cryptocurrency, storage management, decentralized governance, among others.

Valid transactions are maintained in a distributed 'ledger' in the form of hashed information referred to as 'blocks'. Consensus is represented through the longest available chain of blocks that can be obtained from another DLT peer.

The validation of transactions, and therefore the inclusion into the (distributed) ledger, is realized through the consensus layer, realizing computational operations, such as Proof-of-Work, Proof-of-Stake, and others. There has been much discussion on the implications of those computational aspects, e.g., on energy consumption, which is not the focus of this draft.

Figure 1 provides an overview of a typical layering within a DLT architecture. The focus of this draft is on the layers below the session, i.e. the communication that needs to be upheld in order to facilitate transactions and block exchange within the DLT system.

```
+------------++---------------------------------------------------------
| Application||   User    |  DLT   |   DLT    |     DLT   |Decentralized
|   Layer    || Interface | Wallet | Explorer | Analytics |  Finance
+------------++---------------------------------------------------------
|App Protocol||  Identity | Token  |  Storage |    DLT    |Decentralized
|   Layer    ||    Mgmt   |  Mgmt  |   Mgmt   |   Oracle  |  Governance
+------------++-----------------------------+---------------------------
|  Contract  ||         Transaction         |           Smart
|   Layer    ||           Engine            |          Contract
+------------++-----------------------------+---------------------------
|  Consensus ||                 PoW/PoS/DPoS/PBFT/Raft/etc.
|   Layer    ||
+------------++-------------------+------------------+------------------
|  Session   ||    Transaction    |      Block       |      Account
|   Layer    ||                   |                  |
+------------++-------------------+------------------+------------------
|  Transport ||        TCP        |       QUIC       |       UDP
|   Layer    ||      (+TLS)       |                  |
+------------++-------------------+------------------+------------------
|  Network   ||    (DNS + ) IP    |     Service      |     Pub/sub
|   Layer    ||                   |     Routing      |     overlay
+------------++-------------------+------------------+------------------
|  Resource  ||       CPU         |     Storage      |    Transport
|   Layer    ||                   |                  |     Network
+------------++-------------------+------------------+------------------
```

Figure 1: DLT Conceptual Architecture [IIC_whitepaper]

4. Communication in a DLT

With our focus on the communication impact of DLTs, we now tease apart the communication as it usually takes place in a DLT in order to realize the transactions within a distributed ledger and the maintenance of the latter. We first outline the interactions at a higher level before delving into the communication patterns that result from those.

As stated in the introduction, these insights are currently limited to those obtained from Ethereum, a proof-of-work based DLT platform. Future draft revisions will enrich this section with any differing insights from other DLT realizations and platforms.

4.1. DLT Interactions

We can distinguish three core interactions in a DLT:

1. A client commits a transaction to the DLT. The transaction request is being diffused across a set of DLT miners, which

respond to the transaction request separately and add the transaction to their internal ledger information. The commit of the transaction leads to the miners committing compute and storage resources in relation to the smart contract that underlies the transaction. For this, so-called 'proofs' will be executed as part of the computational part of the DLT, although some methods for proof require additional communication to take place, e.g., election protocols.

2. The result of the aforementioned proof is a 'block' (of ledger information) that the miners in turn commit to a set of (other) DLT miners, which each receiving miner adds to their internal blockchain.

3. A client may query the latest blockchain, again from a set of miners to which the query is being sent. The longest returned blockchain represents the most trustworthy ledger information available.

We can see from those interactions above that communication in a DLT is multipoint in nature, i.e., transactions or information (such as blocks) are sent to a set of DLT peers, not just a single one.

Important here is that the set of DLT peers is a randomized sample from a larger pool of available DLT peers; this is to achieve diffusion among many DLT peers, avoiding repeated communication with a fixed set of DLT peers and thereby reducing the threat of collusion of information through a malicious set of DLT peers.

The consequence of that varying random nature of the multipoint diffusion, however, is that repeated unicast replication is utilized instead of efficient network-level multicast; this constitutes a first recognizable impact on provider networks.

In the following subsection, we now focus on the communication patterns that are utilized to achieve the aforementioned interaction. Special attention is here given to the establishment of the pool of DLT peers that is used in the multipoint operations that are executed for each interaction, be it a transaction or the commitment of a newfound (ledger) block.

4.2. Resulting Communication Patterns

As mentioned before, it is key for any DLT peer, be it a client or a miner, to establish and maintain a 'pool of peers' from which it can select a set of DLT peers for each intended interaction. Figure 2 outlines those steps, detailed in the following. Our insights on realization were obtained from an Ethereum based experiment, using the go-ethereum release V1.10.2-stable on a Linux-based machine, operating out of Munich, Germany.

1. The first phase is that of a 'peer discovery'. For this, an initial list of DLT peer information is obtained from a 'bootstrap node', of which only few exist in the DLT, holding the IP address and port information of each DLT peer that has signed up to the DLT overlay (other information may include DLT-specific information, such as an overlay ID or similar).

2. This initial list of DLT peers is now contacted through a (UDP-level) PING/PONG sequence, thereby discovering those DLT peers that are reachable for the DLT interactions.

3. A successful discovery of the DLT peer is now followed with the establishment of suitable transport security. Once successfully secured, the discovered DLT peer is being added to the 'DLT pool' list at the initiating DLT peer.

4. Once security is established, capabilities are exchanged that ensure that the discovered peer can successfully complete possible requests. Those capabilities may include HW capabilities (e.g., GPU usage, certain memory build-out), SW capabilities (use of certain hash functions, blockchain checkpoint) and others.

5. The initiating DLT peer repeats now the previous steps 1 through 4 until the pool size reaches a defined limit. Unlike contacting the bootstrap nodes, however, the newly and successfully discovered DLT peers in the previous round are contacted instead for obtaining a list of DLT peers.

6. Any member of the DLT pool is continuously checked for connectivity through frequent (e.g., TCP-based) HELLO messages. Any failed HELLO transaction leads to removing the DLT peer from the pool and obtaining another DLT peer as replacement.

The final size of the pool is a matter of local configuration (in our case about 28k DLT peers, significantly less than the size of the overall DLT network, which was about 500k at the time of the experiment).

Also, a DLT client may commence with transactions (to the DLT overlay) already while the pool creation is still ongoing, thereby progressing to the last step in Figure 2 once a suitable set of DLT peers can be obtained from the overall (and possibly still growing) local pool of peers.

```
+-------------------+                                  if DLT peer conne
|    Obtain list    |<--------------------------------------+
|   of DLT peers    |<--+                                   |
+-------------------+   | if pool size       +--------------+---
|       Node        |   | smaller than max   |  Maintain peer  |
|      discovery    |   |                    |  connectivity   |
+-------------------+   |                    +-----------------+
|     Transport     |   |
|      security     |   |
+-------------------+   |
|    Capability     +---+
|     exchange      |
+-------------------+
          |
          |   add discovered peers to pool of DLT peers
         \|/
+--------------------------------+
|     Obtain set of DLT peers    |
|     from pool of DLT peers     |
+--------------------------------+
|         Transactions           |
+--------------------------------+
```

Figure 2: Steps of Communications in a DLT

5. Challenges for Users and Provider Networks

Considering the observed communication patterns in the previous section, we can identify a number of challenges that need addressing:

1. Reachability information is required to interact with other peers. For that, bootstrap nodes maintain IP addresses of all peers (plus port information). As illustrated in Figure 2, new DLT peers need to download and expand suitable reachability information upon joining, either from bootstrap node or via discovered nodes - see Figure 2, , requiring each DLT peer to maintain a pool of peer as active connections.

2. Clients know nothing about capabilities of peers to serve requests. In other words, the discovery in Figure 2 merely ensures possible reachability but not necessarily successful communication. As a consequence, the resulting approach, illustrated in Figure 2, is to (1) contact potential peer, (2) wait for connection, (3) inquire capabilities, (4) disconnect if not matching. Here, peers may never reply to connection establishment (step 2), usually resulting in additional latency

due to timeouts involved, prolonging therefore the establishment of the pool of peers to communicate with. Such capabilities often reflect the continuous evolution of business models over DLT networks and may be dynamic in nature. For example, the minimum transaction fee may depend on the 'DLT gas price', which is set up at the transaction recipient (miner).

3. Peers map sending of transactions onto unicast communication, which negatively impacts efficiency (bandwidth usage) and transaction completion time. Here, the use of group-based multicast approaches is difficult due to the random nature of the set of peers selected for communication in every request exchange, aiming at the diffusion of requests rather than interacting with a stable (but possibly colluding) set of peers.

4. DLT peers need to expose their IP address to the DLT system, replicated to the bootstrap nodes, but also other DLT peers by virtue of the discovery process outlined in Figure 2. This may lead to privacy and/or security issues in the form of geo-identifying specific peers, DoS attacks on particular parts of the DLT and others.

6. Experimental Insights

To shed some more light onto the possible impact on provider networks, stemming from some of the challenges in Section 5, we conducted experiments, using the same setup described in Section 4.2. More details (and suitable graphical representations of our initial results can be found in [IIC_whitepaper]).

Here, the goal was to undergo the steps needed to build up the needed pool of DLT peers, after which we sought to synchronize to determine the longest blockchain available in the discovered pool. The resulting geographic spread of the discovered DLT peers included all continents albeit with an expected clustering of nodes North America, Europe, Asia, and Australia, with only few discovered in South America and Africa.

6.1. Types of DLT Peers

Our first target was to differentiate types of DLT peers that stem from the communication patterns in Figure 2. Specifically, we came to differentiate the following types of DLT peers:

1. Non routable peers: This type include all those peers that never positively responded to step 1 of the discovery, i.e. the PING/PONG to determine reachability. Reasons here may include to be located behind a firewall, being intermittently available (and switched off during the connection attempt), or simply

having left the DLT while still remaining in the information pool maintained at the bootstrap nodes.

2. Signalling peers: This type includes peers that respond positively to reachability but do not positively succeed in the transport security or capability exchange steps (blockchain checkpoint).

3. Dropped data peers: This type of peers successfully complete all discovery steps, thereby end up in the pool of peers, but still do not provide suitable data upon request (here a valid blockchain information). The reasons here could be unavailable information or not completing the transfer of information (blockchain information can be very large, several GBs, so that DLT peers may run out of available BW budget or decide to sever the connection because of switch-off or other reasons during the transfer). While here communication in the DLT does take place, it is not successful in regards to the intended communication, therefore wasted.

4. Data peers: This final type of peers successfully completes all steps in Figure 2, i.e. not only the discovery but also the intended transfer of DLT-relevant data.

In our experiments, we determined at about 18% of peers are of the last type, i.e. successfully contribute to DLT purposes, while about 2% are of the third category, about 12% are non routable peers and about 68% are signalling peers. In other words, almost 80% of all attempted discoveries fails either because of the lack of reachability or mismatching capabilities.

6.2. Communication Waste

Looking at the bandwidth usage across the different peer types allows for shedding some light on the communication that needs to be carried through the participating provider networks.

Given the amount of data for each blockchain synchronization, it is not surprising that, despite forming a mere 18% of peers, the 'data peers' account for about 58% of traffic in the overall system. This is followed by the 'dropped data peers' with about 31.5% (since still much data is sent albeit unsuccessfully). Both non routable and signalling peers account for a total of slightly under 10% of data used.

Although the amount of data that is wasted here accounts for (significant) total of about 42%, the data-heavy operation of synchronization large amounts of (blockchain) data is mainly to blame for this; however, the synchronization has to happen for any

DLT peer to start operating as a possible DLT miner, so is not avoidable.

7. Opportunities for Network Innovations

The challenges outlined in Section 5 lead us to outline possible opportunities for network innovations that may address those challenges and reduce the observed impact on provider networks. We stress here that none of the suggested approaches constitute solutions for those opportunities but merely possible starting points beyond which further study is required:

1. Addressing model: With the DLT overlay being realized over an IP network, each DLT peer is being addressed via its IP(v4/v6) address. With the discovery step selecting a dedicated DLT peer (through its IP address), the discovery steps (see Figure 2) include dedicated steps to ensure the reachability of the specific DLT peer under discovery. Until reachability can be ensured, traffic (in the form of PING/PONG messages) and latency (through sending those messages, while needing to wait for a timeout in case the DLT peer is not routable) need to occur, despite the DLT peer not being eventually used for communication.

   *Approaches such as those in [SOI][SarNet2021] [IFIPNetworking2022] may allow for DLT peers to advertise their capability to serve as a miner by using 'service announcements' that expose the capability to serve transaction requests, which each announced DLT peer representing a service instance of the announced mining service. Such native L3 (or L3.5) level service routing capability would therefore remove any of the discovery steps and the maintenance of the dedicated DLT overlay infrastructure. Furthermore, it would remove any visibility of individual DLT peers' reachability information from other miners, until directly communicating with a specific DLT peer (for which the peer's IP address may be used, as suggested in [SarNet2021][IFIPNetworking2022]). Last but not least, being able to send a request without previously forming a pool of DLT peers (which is smaller than the number of all DLT peers in the overlay) also increases the possible number of DLT peers to communicate with rather than being limited to the peer-specific pool.

2. Constraint-based peer selection: Following on the aspect of relying purely on reachability information in the form of IP addresses, the discovery steps in Figure 2 further include a number of capability-dependent selection criteria to finally include a DLT peer in its pool of peers. Specifically, the

security and capability exchange may lead to a disconnect from a successfully contacted DLT because of such exchange leading to mismatching capabilities. Furthermore, even after an initial capability exchange being successful, the actual transaction itself may be constrained by capabilities such as available resources (e.g., bandwidth or CPU), leading to unsuccessful communication, which in turn will need to be compensated with including another DLT peer into the diffusion request.

- Approaches such as [SarNet2021][IFIPNetworking2022] may allow to constrain the forwarding to one of possible many DLT peers. Hence, the capabilities used in the current DLT steps Figure 2 could be encoded as suitable constraints for such selection, the constraints itself being advertised as part of the service announcement (see above). As a result, the request will be forwarded to those destinations only which have previously announced constraints that match those of the request, thereby ensuring the successful completion of the request - further study is needed for those situations in which constraints may change frequently, thereby leading to successful matching, yet still unsuccessful request completion.

3. Diffusion multicast: The multipoint replication of the transaction request to a set of DLT peers, chosen from the larger DLT pool maintained at the initiating DLT peer, increases the overall system but, in particular, individual client bandwidth usage, which in turn impacts the provider network by needing to provide the necessary resources for the replicated sending.

- Approaches such as those in [SOI][SarNet2021][IFIPNetworking2022] may allow for sending a service request to a given number of DLT peers, where the replication is part of the constraint-based forwarding decision, thereby optimizing the packet delivery through in-network instead of endpoint-based replication. The challenge here lies in preserving the diffusion character of the multipoint operation. In other words, the set of DLT peers used for the transactions changes for each request with a randomization that attempts to prevent possible collusion through DLT peers. With that, typical group-based methods, most notably IP multicast, do not suffice.

8. Relation to IETF/IRTF and IEEE SA Efforts

Both, DLTs as well as routing innovations, are subject to investigation in a number of related IETF and IRTF efforts. For instance, the Decentralized Internet Infrastructure RG [DINRGref] has been studying various aspects of DLTs and blockchains. Our findings in this draft may provide additional input into the work of this RG, while we would solicit feedback from this group of experts into the specific insights we have derived so far.

There is no standard way of providing interoperability between DLT networks. This results in difficulty of transferring or exchanging virtual assets from one DLT network to another. An interoperability architecture is being proposed in the IETF [I-D.hardjono-blockchain-interop-arch] to permit two gateways, belonging to distinct DLT networks, to conduct a virtual asset transfer between them while ensuring the asset does not exist simultaneously on both networks. The Open Digital Asset Protocol (ODAP) [I-D.hargreaves-odap] is a gateway-to-gateway protocol to perform a unidirectional transfer of a virtual asset.

Blockchain technologies, and thereby DLTs, have also been proposed for use in network functions itself. For instance, the work in [I-D.mcbride-rtgwg-bgp-blockchain] proposes to position BGP as DLT-managed transactions and thus, to utilize the power of a permissionless (DLT-based) management infrastructure to improve on resilience and trust into the operations performed within BGP, such as origin announcements and BGP updates. Such proposition, however, opens the question on the exact nature of such infrastructure but also its impact in terms of incurred traffic, particularly when operating at scale.

Furthermore, routing innovations under the label of 'semantic routing' have been the topic of recent work, see [I-D.farrel-irtf-introduction-to-semantic-routing] for an overview. With the examples of service routing as possible approaches to realize the opportunities outlined in the previous subsection, a stronger linkage to this activity should be considered.

While the DLT standardization efforts in IEEE SA mainly focus on the upper layers of the DLT architecture, the decentralized identity related standards (e.g., P2958 [P2958] and P3210 [P3210]) that are currently under development might be relevant for addressing specific challenges in the DLT network layer.

9. Open Questions

The work initially presented in [IIC_whitepaper] focused on the specific impact that DLT operations may have on provider networks, thereby turning the attention not to the specific applications of DLT but what their realization may mean to the underlying network operators.

Although attempting from the onset to base our insights on actual experiments we conducted, we recognize that those insights are only the start to a possibly wider understanding beyond this initial work.

We therefore solicit not only feedback on the specific findings presented in the previous sections, but also to specific questions that our work has led to:

1. Correctness of observed DLT behaviour: Is our observed behaviour correct or have we overlooked important aspects?

2. Depth of insights: Can we deepen our insights through more experiments, focus on different or more KPIs?

3. Transfer of insights: Our insights so far are based on the Ethereum DLT system. How transferable are the observed patterns of communication onto other DLT systems that are in use?

4. Differences in DLT realizations: If the answer to the previous question leads to little transfer onto other DLT platform, can we distil those difference with the goal to develop a wider methodology to capture DLT behaviour?

5. Applicability of other network innovations: What other network innovations may address the specific impacts we identified in our study? Which ones beyond the ones currently listed should be included?

10. Next Steps

As for the next steps for this draft, the authors seek to deepen the current insights through further conducted experiments, providing more insights on the disconnects experienced by the system and the costs for maintaining the pool of DLT peers.

Furthermore, the authors will more directly link to relevant network innovations, particularly in the service routing and instantaneous multicast domain, with the goal of providing estimates of improving on the operational costs of DLTs through such new network innovations.

11. Conclusions

This draft is a living document, originating from an initial study in the impact of DLTs on provider networks [IIC_whitepaper].

As such, the authors solicit feedback from the wider DLT and network community to improve on the insights, transfer them onto more DLT systems, and shed light onto how possible network innovations could improve on the identified issues.

12. Security Considerations

This document does not introduce or modify any security mechanisms. The nature of DLTs is to provide a high level of transactional security through immutability of the data in blocks. But 51% attacks are possible amongst miners particularly on smaller, private blockchains where legitimate miners could be prevented from completing blocks and new blocks could be created by illegitimate miners. Smart contracts could become vulnerable if a function calls the wrong contract either intentionally or through human error. Transactional data meant to be private might be exposed. DLT attacks most often involve accounts being hacked outside of the DLT domain.

13. Privacy Considerations

Since the IP addresses of DLT peers are exposed in the DLT system, the DLT network layer might be subject to privacy leakage. This document does not introduce any mechanisms for protecting IP address privacy and the methods described in [I-D.ip-address-privacy-considerations] could be employed to enhance the privacy of DLT peers.

14. IANA Considerations

This draft does not request any IANA action.

15. Acknowledgements

This draft acknowledges the work done in the IIC Industrial Digital Ledger focus group, leading to the whitepaper in [IIC_whitepaper]. We would like to thank the co-authors of this whitepaper for their work, specifically David Guzman (Huawei Technologies), Abhijeet Kelkar (GEOOWN Consulting), Xinxin Fan (IoTex), Mike McBride (Futurewei Technologies), Lei Zhang (iExec), Ulrich Graf (Huawei Technologies) and Dirk Trossen (Huawei Technologies) but also Stephen Mellor (IIC staff) who oversaw the process of organizing the contributions.

16. Informative References

[DINRGref] "Decentralized Internet Infrastructure (dinrg)", WG DIN Research Group, <https://irtf.org/dinrg>.

[DLT_intro] Antonopoulos, A. M., "Mastering Bitcoin, 2nd Edition", Book O'Reilly Media, Inc., 2017, <https://www.iiconsortium.org/pdf/2022-01-10-Impact-of-Distributed-Ledgers-on-Provider-Networks.pdf>.

[DLT_intro2] Rauchs, M., Glidden, A., Gordon, B., Pieters, G., Recanatini, M., Rostand, F., Vagneur, K., and B. Zhang, "Distributed Ledger Technology Systems: A Conceptual Framework", Report Cambridge Centre for Alternative Finance, 2017, <https://www.jbs.cam.ac.uk/wp-content/uploads/2020/08/2018-10-26-conceptualising-dlt-systems.pdf>.

[I-D.farrel-irtf-introduction-to-semantic-routing] Farrel, A. and D. King, "An Introduction to Semantic Routing", Work in Progress, Internet-Draft, draft-farrel-irtf-introduction-to-semantic-routing-04, 25 April 2022, <https://datatracker.ietf.org/api/v1/doc/document/draft-farrel-irtf-introduction-to-semantic-routing/>.

[I-D.hardjono-blockchain-interop-arch] Hardjono, T., Hargreaves, M., Smith, N., and V. Ramakrishna, "Interoperability Architecture for DLT Gateways", Work in Progress, Internet-Draft, draft-hardjono-blockchain-interop-arch-03, 7 November 2021, <https://datatracker.ietf.org/api/v1/doc/document/draft-hardjono-blockchain-interop-arch/>.

[I-D.hargreaves-odap] Hargreaves, M., Hardjono, T., and R. Belchior, "Open Digital Asset Protocol", Work in Progress, Internet-Draft, draft-hargreaves-odap-03, 7 November 2021, <https://datatracker.ietf.org/api/v1/doc/document/draft-hargreaves-odap/>.

[I-D.ip-address-privacy-considerations] Finkel, M., Lassey, B., Iannone, L., and J. B. Chen, "IP Address Privacy Considerations", Work in Progress, Internet-Draft, draft-ip-address-privacy-considerations-03, 10 January 2022, <https://datatracker.ietf.org/api/v1/doc/document/draft-ip-address-privacy-considerations/>.

[I-D.mcbride-rtgwg-bgp-blockchain] McBride, M., Trossen, D., and D. Guyman, "BGP Blockchain", Work in Progress, Internet-Draft, draft-mcbride-rtgwg-bgp-blockchain-01, 29 June 2022, <https://datatracker.ietf.org/api/v1/doc/document/draft-mcbride-rtgwg-bgp-blockchain/>.

[IFIPNetworking2022] Khandaker, K. S., Trossen, D., Khalili, R., Despotovic, Z., Hecker, A., and G. Carle, "CArDS: Dealing a New Hand in Reducing Service Request Completion Times", Paper IFIP Networking, 2022.

[IIC_whitepaper] Trossen, D., Guzman, D., Kelkar, A., Fan, X., McBride, M., Zhang, L., and U. Graf, "Impact of Distributed Ledgers on Provider Networks", Whitepaper Industry IoT Consortium Whitepaper, 2022, <https://www.iiconsortium.org/pdf/2022-01-10-Impact-of-Distributed-Ledgers-on-Provider-Networks.pdf>.

[P2958] "P2958: Standard for a Decentralized Identity and Access Management Framework for Internet of Things", Standard IEEE Standards Association., <https://standards.ieee.org/ieee/2958/10483/>.

[P3210] "P3210: Standard for Blockchain-based Digital Identity System Framework", Standard IEEE Standards Association., <https://standards.ieee.org/ieee/3210/10242/>.

[SarNet2021] Glebke, R., Trossen, D., Kunze, I., Lou, Z., Rueth, J., Stoffers, M., and K. Wehrle, "Service-based Forwarding via Programmable Dataplanes", Paper 1st Intl Workshop on Semantic Addressing and Routing for Future Networks, 2021.

[SOI] Jiang, S., Li, G., and B. Carpenter, "A New Approach to a Service Oriented Internet Protocol", Paper IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2020.

Authors' Addresses

Dirk Trossen
Huawei Technologies
Munich
Germany

Email: dirk.trossen@huawei.com

David Guzman
Huawei Technologies
Munich
Germany

Email: david.guzman@huawei.com

Mike Mc Bride
FutureWei

Email: michael.mcbride@futurewei.com

Xinxin Fan

IoTeX

Email: xinxin@iotex.io