Angieri, S.,García Martínez, A., Liu, B., Yan, Z., Wang, C. y Bagnulo, M. (2020). A Distributed Autonomous Organization for Internet address management. *IEEE Transactions on Engineering Management*, 67(4), pp. 1459 - 1475.

DOI: https://doi.org/10.1109/TEM.2019.2924737

A Distributed Autonomous Organization for Internet address management

Stefano Angieri[1], Alberto García-Martínez[1], Bingyang Liu[2], Zhiwei Yan[3], Chuang Wang[2] and Marcelo Bagnulo[1]

[1]Universidad Carlos III de Mardid
[2]Huawei
[3]China Internet Network Information Center

ABSTRACT

The current system to manage the global pool of IP addresses is centralized in five transnational organizations, the Regional Internet Registries (RIRs). Each of them manages the address pool for many countries. In this paper we present InBlock, a Distributed Autonomous Organization that provides decentralized management of IP addresses. InBlock also fulfills the same objectives as the current IP address allocation organizations, i.e., uniqueness, fairness, conservation, aggregation, registration and minimized overhead. InBlock is implemented as a set of blockchain's smart contracts in Ethereum and it implements all the functions needed for the management of a global pool of addresses without any human intervention. Any entity may request an allocation of addresses to the InBlock registry by performing a (crypto)currency transfer to the InBlock. The fee required, along with the annual renewal fee, serves as a mechanism to deter stockpiling and other wasteful practices.

As with any novel technology, there are many open questions about the usage of blockchains to build an IP address registry. For this reason, we believe that practical experimentation is in order to have hands-on experiences, so we propose to conduct an experiment on InBlock as a starting point to inform future directions in this space.

I. INTRODUCTION

IP addresses are a cornerstone of the Internet. Every device must configure an IP address in order to be able to send and receive packets through the Internet. As such, the management of the global pool of IP address is of uttermost importance for the correct functioning of the Internet. IP addresses are globally administered by the Internet Corporation for Assigned Names and Numbers (ICANN), a private non-profit organization. The global pool of IP addresses is managed through a hierarchical structure rooted in ICANN. ICANN delegates ranges of IP addresses to five Regional Internet Registries (RIRs), the second tier of the hierarchy. RIRs allocate address blocks to Local Internet Registries, (LIRs, usually Internet

Bingyang Liu: Corresponding Author, email: liubingyang@huawei.com

Service Providers). End-users normally obtain addresses from the LIRs and in some cases, directly from the RIRs. In some cases, an intermediate agent between the RIR and the LIR exists, a National Internet Registry (NIR), to accommodate the specific Internet resource management needs of a given country.

While this arrangement has generally been successful, it comes with some rough edges. In particular, one recurring concern is that this structure for the management of the global pool of IP addresses results in the jurisdiction of some countries (where ICANN and the RIRs are hosted) overflowing into all the other countries served. ICANN and the RIRs are private organizations that operate within the legal framework of the countries where they are based. However, they each manage the IP addresses for a large set of countries. This implies that for most countries in the world, the management of a critical Internet resource such as IP addresses operates under the legal framework of a foreign country. This also implies that, for most countries, any legal action involving IP address management will be settled in a foreign court of law, making these resources *de facto* subject to laws of a foreign state, as observed in [1]. These concerns have been recently exacerbated by the development of new Internet security techniques, as we describe next.

Over the last few decades, the Internet has become part of the critical infrastructure for most countries. The increasing concern to guarantee its availability has resulted in the design, deployment and adoption of new security tools, such as the Resource Public Key Infrastructure (RPKI) and BGP Security (BGPsec). These tools aim to provide cryptographic guarantees that whoever is claiming to have an Internet addressing resource is indeed the legitimate holder of the resource according to the defined allocation rules, preventing prefix hijacking attacks and other vulnerabilities. As such, these mechanisms provide the entities up in the hierarchy of the allocation system (i.e., RIRs, NIRs and LIRs) a capability that they lacked so far, namely the capacity to actually enforce the allocations in real time. In particular, they allow entities up in the allocation hierarchy to arbitrarily override an existing IP allocation [2]. So, if/when these cryptographic techniques are widely adopted, the Internet Registries will be able to invalidate allocations, disconnecting whole networks from the Internet, if so dictated by their governing bodies. We note again that a mismatch exists between the geographical scope in which legal, operational and management decisions are taken (the countries where the RIRs are based) and their effects (the whole world). This situation has raised a number of concerns and it may be one of the reasons behind the lag in the adoption of such technologies. Note that the attacks they are designed to prevent are very real, so security measures to protect the Internet are indeed needed.

The current hierarchical design for managing Internet addresses was probably the most natural one when it was created. IP addresses come from a single global pool and in order to properly perform its function, global uniqueness must be guaranteed, i.e., the system must prevent that the same address is simultaneously allocated to two different parties. When the Internet was designed, the straightforward way of accomplishing this was to rely on a hierarchical structure of organizations to manage the allocations of IP addresses, preventing the allocation of the same resource twice.

However, new opportunities for managing namespaces (in our case, the IP address namespace) surface with the recent introduction of the blockchain technology. Blockchains are distributed databases that are controlled through consensus. By design, blockchains are politically and architecturally decentralized [3], i.e., there is no single entity controlling it, and there is no single point of failure in the infrastructure. As

such they prevent any central authority to modify the content of the database. The blockchain technology provides an opportunity to explore alternative IP address management approaches. Moreover, the second generation of blockchain technologies (e.g., Ethereum) allow us to take this approach one step further and explore the possibility to build an autonomous organization that performs the registry functions without any human intervention. The fundamental challenge is how to design the system so that it autonomously performs the registry functions without being subject to abuses and misuses from the (human) users.

In this paper, we present InBlock, a Distributed Autonomous Organization for managing IP addresses. The proposed approach uses blockchain technology to perform IPv6 address registry functions. While InBlock supports both IPv4 and IPv6, we focus on IPv6 as IPv6 has a large remaining pool of unassigned addresses, while the vast majority of the IPv4 address space has already been assigned [4]. InBlock makes different trade-offs than the current hierarchical allocation system. First and most importantly, InBlock is not centrally controlled, but as any blockchain-based mechanism, it depends on distributed consensus. Second, InBlock operates completely autonomously, without any human intervention. InBlock provides a distributed, automatic, irrevocable, tamper-free, publicly accessible and privacy-preserving resource allocation mechanism, designed with the appropriate (economic) incentives to enforce address conservation. We show that the proposed InBlock design is able to perform the IPv6 registry functions in an efficient manner, significantly reducing the operational costs compared to a traditional (human-based) registry and also reducing in orders of magnitude the time required to perform an allocation. In addition, this solution is compatible with the cryptographic security architecture developed for the Internet routing system.

InBlock is a set of programmes that autonomously runs in the blockchain, performing the functions of an IP address registry, as defined by [5]. In a nutshell, InBlock works as follows: InBlock has a block of globally routable IP addresses to allocate. To request an address allocation, an entity transfers a fee to InBlock. Once the fee transaction has been verified, InBlock annotates in the blockchain the prefix allocated to the requester, serving as a ledger of the address assignment for any interested party accessing to the blockchain. Note that, following the paradigm of *code is law*, the organization bylaws are defined in the InBlock code and executed in the same way by any node validating the blockchain. The human action is limited to the initial definition of the allocation rules (before the InBlock is executed in the blockchain). As a blockchain-based organization, it is not controlled by any single entity, so when applied to IP address management, it results in a mechanism that reflects more accurately the current Internet reality as a global network. This approach provides clear and transparent rules without any kind of human discretion, and thus, it reduces legal uncertainty costs for the organizations depending on the Internet for their activities. Address conservation is preserved by a fee mechanism which mimics the current fees charged by the RIRs to the recipients of IP address allocations. In addition, InBlock provides the means to become the authoritative database in which the assignee of the prefix can associate the basic information for the Internet routing system to operate, currently stored in the Internet Routing Registries [6], along with the cryptographic information that may be used to secure the routing system.

It is worth to note that InBlock is designed as an additional IPv6 registry and not as a replacement for the current IANA/RIR based one. The current hierarchical system has stood the test of time, and the ICANN plus RIR system provides services that are appreciated by the Internet community. Therefore,

we do not devise a migration strategy in which the information currently hold by the RIRs is transferred to a blockchain based mechanism. We envision InBlock as just an alternative for organizations deeply concerned about the mismatch in jurisdictions, including the RIRs themselves, which may not want to be responsible for address allocations that are subject to legal processes either in their jurisdiction of in the area in which they operate.

As it is the case for most new disrupting technologies, there are many open questions about blockchains including how sustainable are they, how they will evolve, how secure are they, among many others. We now lie on a crossroad: we have a technology that has the potential to bring autonomous decentralized IP address management to the Internet, but it is still too immature to be fully adopted. On one hand, it is in the Internet's genetic code to embrace innovation, but on the other hand, there is too much uncertainty about blockchains and too much at stake to simply give a step forward and adopt an Internet-wide Blockchain-based registry.

For the aforementioned reasons, we believe it would be beneficial for the Internet community to perform a series of experiments on decentralized Internet address management using the blockchain technology. The proposal is to allocate a small IP address block out of the global address space for an experiment and to create one or more blockchain-based organizations to manage the allocations out of that address block, with a predefined lifetime. Such an experiment would enlighten the community with a hands-on experience to inform future directions regarding decentralized Internet address management.

The rest of the paper is structured as follows: In section II we describe the current IP address allocation system and the entities involved. In order to understand which are the measures that current key entities can apply to restrict the ability of third parties to communicate, we sketch the basic notions of the Internet routing system, and the cryptographic security standards proposed. In the next section, we introduce the blockchain technology, with emphasis on one of its incarnations, Ethereum, as it is the platform of choice for developing the InBlock solution. Section IV is devoted to present the rationale for the design of InBlock. For doing this, we analyse each of the requirements for an address allocation system, and how they are fulfilled by InBlock. Besides, Ethereum restrictions are analysed, to show that price, latency or throughput of blockchain operations are appropriate for address allocation needs. Then we describe how InBlock behaves identifying the roles of the actors involved and we detail how basic operations proceed. In section VI we present an experiment in which InBlock is used to allocate a small set of addresses. We describe the objectives of the experiment, and how to run it in controlled, yet realistic, conditions. We next describe related work, and we end with the conclusions.

II. Background on IP address management and routing

A. IP Address management

IP addresses identify the end-points of every Internet communication, providing both identity and location functions. An IP address is a 32-bit identifier for IPv4, and a 128-bit identifier for IPv4's intended replacement, IPv6. Each network is assigned one or many ranges of IP addresses (called prefixes), which are non-overlapping with the prefixes assigned to other networks. The administrator of each network then assigns IP addresses to the nodes at the network. The assignment process must consider the limitation of the pools at the time of allocation, and the need to ensure uniqueness and proper registration to meet

several operational requirements [7, 8].

In the early days of the Internet, the responsibility for assigning and managing the global pool of IP addresses was performed by the University of Southern California as part of a research project funded by the Defence Advanced Research Projects Agency (DARPA), U.S. government. Between 1998 and 2016, the National Telecommunications and Information Administration (NTIA), part of the Department of Commerce, U.S. government, outsourced the technical management role to the ICANN, the Internet Corporation for Assigned Names and Numbers. In 2016, Internet stakeholders, including the U.S. government, agreed to let the contract with the U.S. government expire and ICANN continued to perform the management of the global pool of Internet addresses, numbers and names ever since [9]. Although ICANN's structure seeks for accountability and transparency, ICANN is ultimately subject to the jurisdiction of the California State, in which many lawsuits have been filed [10].

ICANN has delegated the address management duties to the five Regional Internet Registries, RIRs, namely AFRINIC (Africa), APNIC (Asia Pacific region), ARIN (mainly US and Canada), LACNIC (Latin America and the Caribbean) and RIPE (Europe, Middle East and Central Asia). RIRs are open membership-based bodies composed primarily of organizations that operate networks. The address resources received from ICANN are assigned according to policies developed regionally by each RIR, although coordinated with the rest. Then, the resources are allocated to their requesters, according to six goals explicitly agreed among all RIRs in its policies [11–15]:

- **Uniqueness**. Addresses must be globally unique, the "raison d´être" of the registry.
- **Fairness**. Current policies are designed to be fair, in the sense that they should be equally applied to all parties irrespectively of "their location, nationality, size, or any other factor" [11].
- **Conservation**. A main goal of the Internet resource allocation policies is to make a rational use of them and avoid wasteful practices.
- **Aggregation**. The core routers of the Internet networks exchange information about the Internet address space assigned to each network to perform the global routing function. In particular, these routers must store and process advertisements for the prefixes that describe the address space assigned to every network, so the advertisement of a route can be accounted as an *externality* [8]. The lower the number of prefixes exchanged, the lower the hardware requirements imposed to all the routers participating in the interdomain routing system. The allocation policies aim to reduce the number of prefixes advertised by fostering hierarchical allocation to some extent. In particular, allocation policies encourage the use of provider-based address aggregation as a preferred choice, by giving large address blocks of so-called Provider Aggregatable (PA) addresses to network providers, which in turn suballocate them to end users. In this way, the routes to many different end users can be advertised by a single announcement that encompasses the address space of the suballocated blocks, thus reducing the number of entries to store and process in the core routers. RIRs may also perform Provider Independent (PI) assignments directly to end-users, usually smaller, although at the risk that they may be unreachable due to network operators not assuming the cost of routing them [16].
- **Registration**. Contact and other information associated with allocations is stored to help the normal operation of the Internet, such as serving to troubleshoot connectivity incidents. The current IANA-RIR based system maintains both a private and a public database with information regarding the

allocations: first, any registry allocating an address block keeps (internal) records on the party receiving the resources. This typically involves a contract which includes detailed contact information. This information is private and it is not used for Internet operations. It can be used for legal purposes as long as the legal actions are valid within the legal context of the host country. In addition, the party obtaining the resources may use any of the available Internet Routing Registry (IRR) to publish contact and technical information related to the resources. However, the information stored in the IRR system is often incomplete or inaccurate [6].

- **Minimized overhead**. The allocation system should work with as little overhead as needed to fulfil its function.

RIRs can allocate PA blocks to LIRs, and they may also provide direct PI assignments directly to end-users. All RIRs define a minimum PA IPv6 allocation of /32 [11–15][1]. The allocations can be (much) larger if the applicant justifies the needs. In particular, the larger allocations so far are /20. The minimum PI allocation is a /48 or a /56 depending on the RIR and they can be larger if justified.

RIRs charge a yearly membership fee to entities holding Internet resources. In all RIRs except for RIPE, fees vary according to the amount of resources received. A /48 PI allocation fee is between US$ 100 and US$ 800, depending on the RIR. A /32 PA allocation fee ranges between US$ 1,000 and US $ 2,500. See figure 1 for the detailed information on the fees.

The current fee structure used by the RIRs is not lineal with the number of addresses. The fee for a /32 is roughly one order of magnitude larger than the fee for a /48 while a /32 contains 2^{16} more subnets and addresses than a /48. A similar effect can be observed in PA allocations of different size, meaning that the fee for a /20 is significantly less than 2^{12} times the fee for a /32.

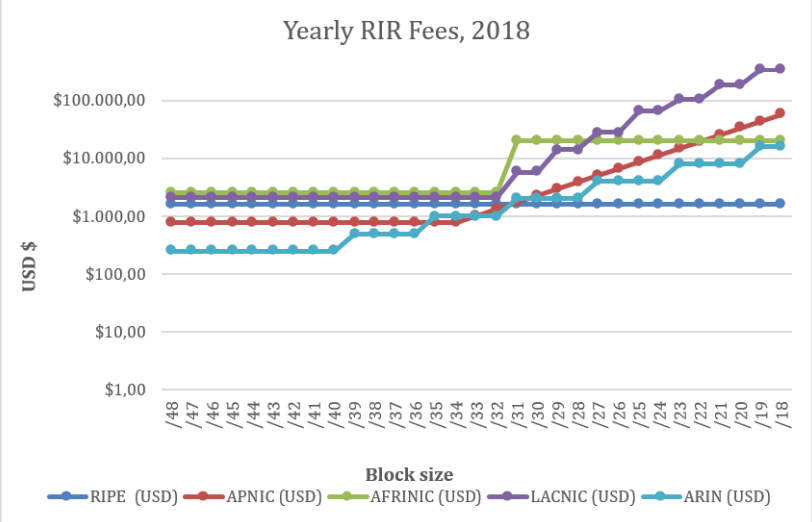

Fig. 1: Yearly RIR Fees for Provider Aggregatable address assignments, 2018

[1]An /32 allocation accounts for $1/2^{32}$ of the address space, and contains up to 2^{32} different IPv6 subnets, as each subnet is a /64 prefix in order to accommodate 64-bit identifiers [17]

B. Interdomain routing

The Border Gateway Protocol (BGP) is used to exchange prefix reachability information between the different networks in the Internet. The function the BGP protocol performs is called *interdomain routing*. The different networks participating in the BGP protocol are identified through AS numbers, which are 32-bit unique identifiers. The AS numbers are managed in a similar way than IP addresses through ICANN and the RIRs.

The original BGP specification lacks of security features, enabling the unwanted manipulation of routing information. For example, an attacker can advertise someone else's prefix as its own, to hijack the traffic for that prefix. In order to prevent such incidents, the BGP ecosystem has been recently enhanced with origin validation capabilities.

Origin validation is provided by the RPKI [18] architecture. The RPKI architecture defines a distributed repository that contains Route Origin Authorizations (ROAs), X.509 certificates that are used to assert that a network, identified by its AS number, is authorized to announce a given prefix BGP (and thus, receive traffic to it). The trust chain of the RPKI starts from the RIRs, that issue certificate delegating prefix ranges to the LIRs, or to the end users, which can in turn issue ROAs. This arrangement is shown in Figure 2, in which RIRs issue certificates to LIRs to authorize to sign ROAs for address blocks, and LIRs can further delegate this authorization to end users.

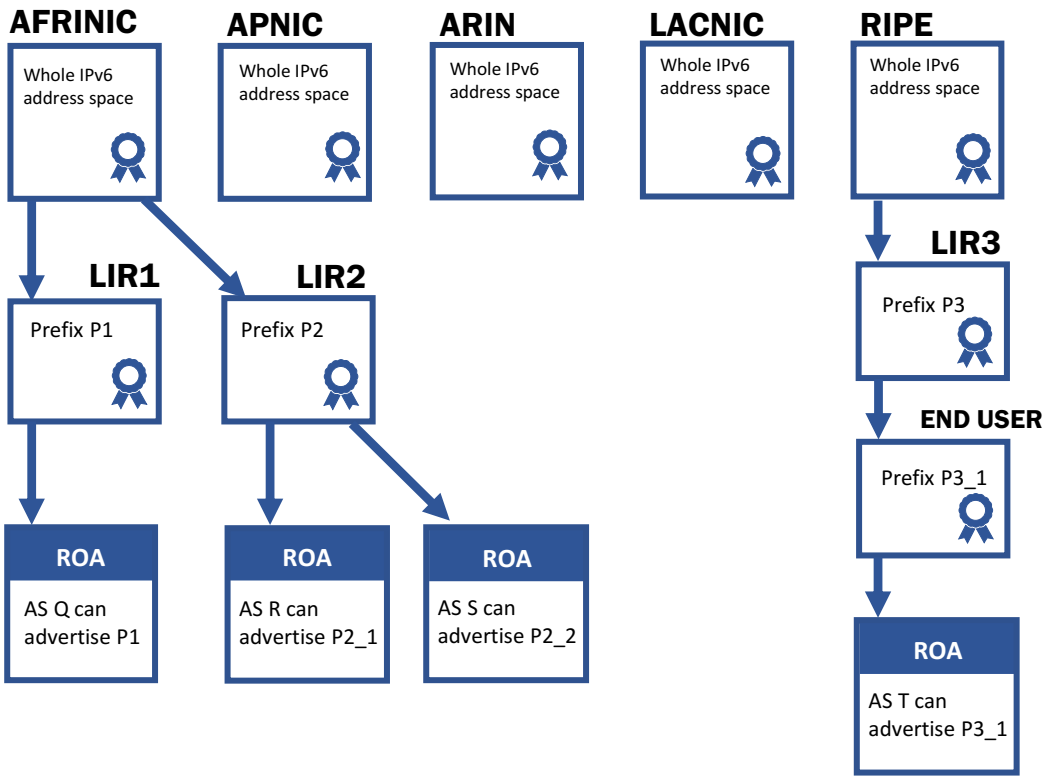

Fig. 2: Hierarchy of the RPKI and ROAs

The RIRs also provide access to repositories with the cryptographic information issued so far, which at the time of this writing, comprise around 14,000 ROAs [19]. ROAs can be used to discard BGP

advertisements, and thus, prevent reachability to certain prefixes, AS numbers, or combinations of both.

Currently, each RIR has its own RPKI root certificate, a self-signed certificate for the whole address space range of IPv4 and IPv6 [20]. A network willing to perform origin validation for the routes to the networks of any region of the world can configure the root certificates of the five RIRs as Trust Anchors, and download the repositories they hold. Then, they use this information to build the router configuration that will filter BGP route information in real time.

[2] shows that certificate issuers can manipulate at will the contents of their publications; for example, they can issue certificates to invalidate or supersede previous resource delegations, affecting connectivity with or through networks performing origin validation. Top-level RPKI entities, i.e., the RIRs, can invalidate or supersede resource allocations they previously allocated, so they can prevent communication for any network with a certificate rooted in the Trust Anchor of the RIR considered.

Moreover, because each RIR currently has a root certificate for the whole IP address space, any RIR can issue valid certificates for any IP prefix. This means that a RIR can issue a certificate for prefix for which another (valid) certificate, rooted on a different RIR, exists. For example, although prefix P was assigned by RIR_A to an entity in its region, out of its own address pool, and a certificate rooted at RIR_A for prefix P exists, RIR_B may issue a conflicting certificate for the same prefix. The resulting behavior of these actions vary for different third-party networks and routers, and will depend on the configuration of the software applying the validation rules over the RPKI information. The network exposed to the conflicting certificates may decide to select one of the certificates (based on explicit preferences on the Trust Anchor, or in the order in which the information has been processed) or discard both. The implications are that the isolated action of a RIR over a prefix can invalidate it for all the networks performing any type of cryptographic validation if the prefix belongs to its region, and can interfere in ways that are hard to predict for prefixes belonging to the address space assigned to other RIRs.

III. Background on Blockchains

A blockchain is an immutable distributed ledger that records validated transactions permanently without the need of a trusted third party.

The blockchain is a distributed ledger because all the information is stored in all the nodes composing the blockchain peer-to-peer network[2].

The blockchain is composed of an append-only list of blocks, securely linked between each other through cryptography [21]. Every block contains a hash pointer to a parent block, a timestamp and transactions' data. The addition of new valid blocks is determined through a distributed consensus mechanism. The consensus is an emerging artefact representing the agreement reached by more than thousands of nodes on the blocks added to the blockchain. The most popular consensus mechanism is Proof-of-Work (PoW) [22].

In PoW, nodes try to solve a complex mathematical problem in order to gain the right to append a block to the existent chain (and make some profit). New block signers are chosen through a *mining* race. Every time a block is added, a new mining race starts and every miner tries to find the solution to gain the next mining block contest and receiving the related fee. Since every miner is working on the same

[2]This is different than other distributed databases, where different parts of the database are stored in different nodes with a limited level of replication just to achieve redundancy and performance benefits

problem, once the challenge is solved by someone, the computational power spent from the other miners on the same problem is wasted. PoW consensus mechanism is expensive in terms of energy consumption.

The hash structure of the blockchain makes computationally unfeasible to alter the data of one block without the manipulation of all subsequent blocks. Tampering the ledger then requires both the collusion of the majority of the network and an enormous amount of computational power to rebuild the chain from the replaced block. This is the sense in which we interpret the immutability of the blockchain.

Finally, we stress that only valid transactions are included in the blocks forming the blockchain. To determine if a transaction is valid, all nodes participating comply with the same block validation rules. As an example of these rules, a value transaction must be signed with the private key of the originator.

The blockchain paradigm can be extended to the automation of complex resource manipulation and transference procedures in a transparent and trustable manner, by means of the specification of *smart contracts*.

A smart contract [23, 24] is a programme that is stored in the blockchain and it is executed by the nodes of the blockchain network. Once deployed in the blockchain, the blockchain nodes will execute the smart contract whenever a monetary transaction to the contract account triggers its execution. Then, every node validating the blockchain will execute the code of the contract, written in the blockchain itself, reaching the same final state.

Ethereum [25] is a public blockchain platform created to facilitate the development of smart contracts. Ethereum has a built-in Turing-complete programming language that allow developers to easily write smart contracts. Every operation in the network is triggered by transactions between accounts, either Externally Owned Accounts (EOAs), owned and controlled by users, or Contract Accounts, associated to a Smart contract which code and state are stored with the account itself. Being a public blockchain, any party can create one or more EOAs and run (or deploy) a smart contract in Ethereum.

Ethereum has implemented a PoW based consensus mechanism. Miners are rewarded in Ether, Ξ, the Ethereum cryptocurrency, for the storage and processing power they contribute to. Ethereum users that want to run a smart contract or to use one, issue a transaction in the Ethereum network which includes a transaction fee payable to the miners. The value (in Ether) of the transaction fee is set by the user generating the transaction and should reflect the number of operation steps to be performed to accomplish a certain work and the priority that the user wants to get from the blockchain miners, as higher transaction fees imply that the transaction will be processed earlier by the miners. Transaction confirmation times are estimated around 10-15 seconds depending on storage needs, code complexity and bandwidth usage.

Ethereum enables the deployment of a Decentralized Autonomous Organization (DAO) [26], an organization that is fully implemented in the form of one or more smart contracts without any human involved in the daily operation of the organization. In particular, the bylaws of the organization are embedded into the code of the smart contracts. DAO's financial transaction records and program rules are maintained on the blockchain.

A. The Blockchains in nowadays society

Blockchains, and in particular, cryptocurrencies, one of the several blockchain-based applications, have already attracted the attention of media, business enterprises and governments [27]. the financial interest

of this technology is beyond question, with a market Cap of US$ 187 billion for the aggregate of the 2,164 (and rising) different cryptocurrencies [28, 29].

However, blockchain appeal exceeds the financial sphere. Blockchains, as a new general purpose technology yet in an early stage of development, exhibit a potential to disrupt everyday life comparable to computers or Internet itself. A wide adoption of blockchain can bring a higher degree of automation, the progressive elimination of intermediaries, an easier and faster money circulation. For example, it has been estimated that Blockchain can led financial institutions to save $20 billion per year in crossborder payment costs, settlement and regulatory [30]. In addition, blockchain come with the promise of a much yearned transparency in the relationship with companies and institutions [31]. We can cite in the area of Corporate Governance the protection of stake-holder interests by means of real-time access to manager actions (including their trading activities) [32].

Blockchain technology opens new innovation opportunities. One of the most disruptive prospect emerges from the combination of the code-ification of law paradigm with blockchain smart contracts. This alliance can empower the ex-ante enforcement of technical rules, despite the difficulty and the cost of transposing legal specifications, written in natural language and so inherently ambiguous, into technical rules based on mathematical models and formal algorithms [33].

In this vein, the application of the blockchain technology to the management of the Internet resources result naturally from the convenience to provide a fair and predictable, transparent, automatic and efficient framework. The InBlock DAO experiment, automating the process of assigning Internet resources to the user complying with the "law" written in its smart contracts, is a first step in the exploration of the blockchain potential in that field.

IV. Description of the proposed InBlock DAO

InBlock is a Decentralized Autonomous Organization that performs IPv6 address allocation registry functions. By autonomous, we mean that the whole organization lies in the blockchain, in the form of smart contracts running in the blockchain without human intervention. It is decentralized because the smart contracts run on the nodes that are part of the blockchain. The behavior of the registry is governed by the code of the smart contract. Modifications to the information regarding the registry of IPv6 address blocks are triggered by blockchain transactions and subject to the consensus mechanism of the blockchain. Therefore, the smart contracts define what a valid transaction is and then all the nodes of the blockchain will enforce that only valid transactions modify the address allocation registry information.

InBlock is configured with a block of globally routable IPv6 addresses to allocate. When an entity wants to obtain an address allocation, it uses its Ethereum account to perform a request. The request is basically a blockchain transaction that transfers a predetermined fee (paid in Ether, the Ethereum cryptocurrency) to InBlock. InBlock verifies that the transaction is valid and that the fee has been correctly transferred. Upon reception of the transaction, the InBlock code goes through its state (stored in the blockchain) and finds an address block that is not currently allocated. Once an available block is found, InBlock associates the block with the identity of the entity requesting the block. This allocation information is recorded in the blockchain.

The allocations have a predefined lifetime. The holder of the resources can renew the allocation making

a new transaction transferring the yearly fee to the InBlock before the expiration date. If this happens, InBlock extends the lifetime of the allocation for another period.

Each IPv6 allocation record stored in the blockchain contains the information about the allocated prefix, the holder Ethereum Identity, the expiration date and a pointer where to find additional information about the holder of the allocation, allowing the holder to include contact or other information.

As stated above, InBlock defines the rules that govern the IP address allocation. Phrased in the Internet Registry jargon, this means that the smart contract will encode the IPv6 address allocation policy. It is only natural then that the goals for the design of the InBlock are aligned with the goals of the existent IPv6 address allocation policies. We next describe the different mechanisms that are part of InBlock aimed to fulfill the address assignment goals of uniqueness, registration, aggregation, conservation, fairness and minimum overhead stated by the current RIR-based allocation system (see Section II). As mentioned earlier, the main challenge is to achieve these goals autonomously without mediating any form of human intervention during the process, given that the InBlock's users will be human that will try to misuse and abuse the system. One major concern is how to deter stock pilling or other forms of address wasteful practices.

A. Uniqueness

To guarantee uniqueness in the address assignment, we first require that the block of globally routable IPv6 addresses assigned to InBlock is unique (reserved exclusively for this purpose). Then we rely on the InBlock code, stored state, and the blockchain consensus mechanism to ensure that only unique assignments are valid, and thus, included in the blockchain registry.

B. Conservation

We identify two different concerns affecting conservation namely stockpiling prevention and the reclaim of unused addresses. We describe the mechanisms used by the InBlock to deal with each of them separately.

1) Stockpiling prevention: One major concern to be considered when designing an IP address registry is how to prevent stockpiling, i.e., the accumulation of resources beyond the actual legitimate needs of the requesting entity [11–13, 15]. IP addresses are valuable assets and given the precedent regarding IPv4 address space exhaustion, some parties may be tempted to obtain IPv6 addresses just in case they become scarce in the future. The InBlock design must provide the means to prevent or at least to control the extent of stockpiling. The current IANA-RIR system *de facto* uses four mechanisms to prevent IPv6 address stockpiling. First, they require a justification for the need of the resources requested, based on planned needs for IP addresses for initial allocations and based on the HD ratio (Host-Density ratio [34]) metrics regarding subsequent allocations. Second, the requirement to become member of the RIR, paying an initial fee, and the charge of a yearly fee to the parties holding resources, in most cases in relation to the amount of resources received (see 1). Third, an abundance argument: because there are enough addresses in the IPv6 global pool for all future needs, there is no point for LIRs and end-users to request addresses for the sake of stockpiling. And fourth, the possibility of reclaiming the addresses allocated if the holder of the resources does not comply with the requirements defined in the allocation policy.

In InBlock, we explicitly give up the first and the fourth mechanisms. In order to achieve full decentralized control, the process of granting a new allocation and the process of renewing an allocation must be fully automatic and encoded in the blockchain. Both the first and the forth aforementioned mechanisms require some form of human intervention, making them incompatible with the InBlock design goals. We argue that the remaining two mechanisms, fees and abundance of addresses, are enough to prevent stockpiling in the IPv6 case.

We next need to define the fee and allocation size structure that suits the InBlock purposes and it is efficient deterring stockpiling. We use as a starting point the current size and fee structure used by the RIRs.

As mentioned in section II, the current fee structure used by the RIRs is not linear with the number of addresses. This poses a challenge for an automated mechanism based in fees to deter address waste such as the one we aim to design for the InBlock. The lower the cost is per address is, the less effective is the mechanism to deter stockpiling and other wasteful practices. On the other hand, if the fee is set to the largest cost per address currently used by the RIRs (e.g., to the cost per address used in /48 PI allocations), this would render the cost of a larger impractically high (the cost of a /32 would be tens of millions of US$ is the cost per address of a /48 is used).

It is challenging for the InBlock to have different cost per address depending on the size of the allocation, because this may encourage applications for larger blocks even when not needed, resulting in address waste (note that we do not have a complementary mechanism such as a need assessment, to modulate user requests). On the same vein, having larger allocations with a low cost per address may promote a secondary market, where it is possible to obtain larger allocation for the same fee (or less) that it would take to obtain a smaller allocation directly from the InBlock[3]. This would again create the incentives for applicants to obtain larger allocation than what they would really need through the secondary market, resulting yet again in address waste.

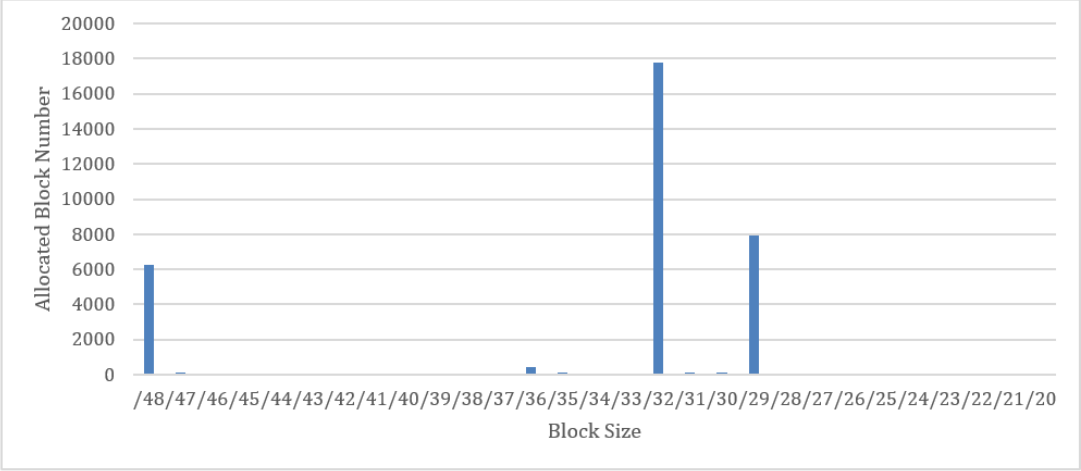

Fig. 3: Distribution of number of blocks allocated up to May 2018, per block size.

[3]Note that this is a risk of the current fee system, in which the cost grows in a sublinear trend with the number of addresses.

In figure 3 we depict the distribution of block sizes of all the existent allocations (for all the RIRs[4]). We observe that, at the time of this writing, there are 17,795 allocations of /32, 6,283 allocations of /48, 7,903 allocations of /29. There are 191 allocations larger than /29. So, roughly half of the existent allocations are /32s, 25% are /29s and the remaining 25% are PI allocations of /48.

Considering all the above, we propose that the InBlock only allocates /32s and /48s, charging a fee similar to the one found in the RIRs. By doing this, we can satisfy the most common allocation sizes. If an entity requires more than a /32, it probably can afford to request 2 or even 4 /32s, paying the corresponding fee for each of them. Since the fee increases linearly with the number of addresses for allocations larger than a /32, we believe that this fee structure would be enough to deter request additional /32 blocks that are not really needed.

This still does not address the problem regarding wasteful allocations involving address needs smaller than /32. Current RIRs fees are such that a /32 costs in the range of US$ 2,500 to US$ 1,000, and a /48 (PI) costs in the range of US$ 100 and US$ 800 (plus an additional initial fee in the range of US$ 250 to US$ 2,500). Suppose that the InBlock charges US$ 2,500 for a /32 and US$ 300 for a /48. Since the fee is one order of magnitude larger for a /32 than for a /48, this is likely to be enough to avoid the majority of applicants that would satisfy their needs with a /48 to request a /32, reducing address waste.

A final argument to support that the proposed fee structure is sufficient to deter stockpiling is the following: If the fee of a /32 is 2,500 US$, then getting the whole IPv6 address space would imply a total amount of $10.5 * 10^{12}$ US$ (compared to the world Gross Domestic Product, GDP which is $80 * 10^{12}$ US$ for 2017 [35]), making it impossible for any party to even get hold of a significant chunk of the IPv6 address space.

Some further considerations about the fees:

- **Currency**. InBlock will define the fee in a fiat currency (Euros, US$, Yuans) but the InBlock will actually collect the fee in the cryptocurrency used in the blockchain (Ether in Ethereum). Because the prices of the cryptocurrencies fluctuate significantly, we propose to define the fee in the fiat currency and convert the fee from the fiat currency to the current correspondence in Ether.
- **Fee Update**. Fees cannot be constant because if the fiat currency value devaluates, the fee will be less effective as a mechanism to deter stockpiling. On the other hand, InBlock is designed to work without human interaction. We propose to link the evolution of the fee to the increase of the world GDP. In this way, the fee will be updated to preserve its value in the future. Also, making the update of the fee automatic (and preventing any human interaction when defining the fees in the future) provides a stable framework for parties holding resources and prevents from having a human deciding on the fees (which may give the human the power to arbitrarily increase the fee, and to push the parties out of the system).
- **Destination of the fee money**. It is worth to note that the fee, in the ranges discussed above, is just a mechanism to prevent stockpiling. The actual cost of running the InBlock is way much less

[4]ftp://ftp.afrinic.net/pub/stats/afrinic/delegated-afrinic-extended-latest
ftp://ftp.apnic.net/pub/stats/apnic/delegated-apnic-extended-latest
ftp://ftp.arin.net/pub/stats/arin/delegated-arin-extended-latest
ftp://ftp.lacnic.net/pub/stats/lacnic/delegated-lacnic-extended-latest
ftp://ftp.ripe.net/pub/stats/ripencc/delegated-ripencc-extended-latest

than the fee (we estimate 15 US$, see IV-G). Several institutions involved in the resources (ICANN, RIRs, IETF, ISOC) could be natural recipients of this money. We note that the actual destination of the money is not relevant for the mechanism, as it would even work if the money were just destroyed (sent to an non-existent account).

2) Reclaiming unused addresses: With the exhaustion of IPv4 addresses, attention has been paid to policies to reclaim unused addresses. However, only market-based approaches have proven to be effective, as coercive measures face many challenges [4, 8]. The InBlock expiration model provides a mechanism to reclaim unused addresses. When an address allocation is not renewed before its expiration date, the address block returns to the pool and the address space is eligible for further reassignment.

Besides, this mechanism serves to fix a well-known issue with storing tokens in blockchains: if the holder of the resource loses the private key that secures the allocation, the resource is lost. In particular, it is estimated that 20% of existing Bitcoins are "lost" because of this [36].

C. Aggregation

Another important consideration that an allocation mechanism needs to address is related to the preservation of the global routing table. Due to the large size of the IPv6 address space, fostering aggregation is crucial for the viability of the routing system. Current RIR allocation policies promote aggregation through the preferred use of PA addresses, but they still allow PI allocations. PI allocations, usually much smaller than PA, contribute to the size of the global routing tables, as they cannot be aggregated. In the previous section we presented a fee structure to deter parties from requesting larger allocations than needed. In this section, we discuss the factors that hinder the deaggregation resulting from a secondary market of address blocks. We also present how InBlock enables aggregation of multiple address blocks obtained by the same entity through a sparse allocation policy.

1) Deagggregation of address blocks due to secondary market transfers: One potential concern with the proposed fee structure is that it may stimulate the creation of a secondary address market that may extend the use of PI allocations by end sites as opposed to PA allocation from the LIRs.

Currently RIRs charge a fee for each PI allocation. While the current policies do impose a number of requirements on end sites requesting PI block, these requirements are usually in the form of a plan or a prevision regarding number of hosts/sites in the near future. We argue that because the InBlock charges a similar fee than the RIR for a PI /48 allocation, it is unlikely that there will be an increased demand of PI allocations due to the InBlock.

The current fee structure would make economically attractive for a party to obtain a /32 from the InBlock and re-sell smaller allocations cheaper than the corresponding fee from the InBlock. This may render PI-sized allocations more affordable, and parties that would not be willing to pay the yearly fee that the RIR or the InBlock charges for a /48 allocations, may be willing to obtain a much cheaper /48 from this secondary market. Since these would indeed be PI allocations (as they are not provided by the ISP), they are likely to be announced in the global routing table as separate routes, bloating the routing table. Note that PI allocations are attractive because they avoid provider lock-in and the associated renumbering cost if the customer changes ISP.

However, a /48 obtained in the secondary market is not a perfect substitute of a /48 obtained directly from the InBlock. The reason is that there is an intermediary in the /48 obtained through the secondary market, the reseller. If the intermediary fails to renew the /32 where the /48 are extracted from (for example, because it goes bankrupt), all the /48 allocations will not be renewed and the end sites will lose their addresses. Moreover, once the end site has obtained an allocation from the intermediary and configured in their network, the intermediary has an incentive to increase the charged fee since there is a cost from the end user to renumber its network. The end site would then be trading ISP lock-in for intermediary lock-in.

Nevertheless, it is still possible that some sites find it attractive to obtain PI blocks in the secondary market, and negatively impacting the global routing table. In order to prevent this, the InBlock could limit the number of different AS numbers used in ROAs within a single /32 to a maximum number (e.g., 100 different AS numbers). This restriction would not impose a real restriction in the operations of ISPs using the /32 for PA allocations, but would negatively affect the intermediaries that want to resell PI allocations out of a /32, since it would prevent all the end sites obtaining an allocation out of a single /32 to use different origin ASes in their ROAs (which is what they would naturally do when announcing a PI block in the interdomain routing).

2) Sparse allocation: In order for the multiple blocks assigned to a single entity to be aggregatable, the InBlock will use a sparse allocation strategy [37] to manage the overall pools. This allows the holder of a resource to request for the contiguous block, so that the multiple blocks that a given entity obtains from the InBlock are aggregatable.

When submitting a new request for a new block, the applicant can attach a proof that it holds a block from a previous allocation. If this is the case, InBlock will allocate a contiguous block, enabling the aggregation of the two prefixes.

D. Registration

In the case of the InBlock, blockchain identities are mostly anonymous[5]. Moreover, payments are done using the corresponding Ether, which as of today provides significant anonymity features. All this implies that the InBlock has no information about the entity that received the allocation. InBlock provides the means to voluntarily include contact information or route policies for each allocation, in a similar way to Internet Routing Registries [6], but it does not mandate it. However, it is worth to note that the blockchain enables a cryptographic link between the holder of the account to which the address prefix has been assigned, and the routing information. This means that any other party accessing to the blockchain can verify that the contact and routing policy information is authorized by the assignee of the Internet resources.

In summary, InBlock provides stronger privacy features than the current IANA-RIR system, while still enabling resource holders to voluntarily provide contact and routing information for operational purposes.

[5]Blockchain identities are not perfectly anonymous, since there are means to try to link an Ethereum identity to a physical entity, but in general, it is not required prior identification to obtain a Ethereum identity.

E. Fairness

Fairness is an explicit goal in current RIR address allocations policies. Fairness in this context means that the policies should be equally applied to all parties irrespectively of "their location, nationality, size, or any other factor" [11]. InBlock naturally achieves that goal, since any party can obtain an Ethereum identity and then it can obtain an allocation from the InBlock.

Moreover, InBlock takes a step further, as it achieves jurisdictional fairness. As stated earlier, one explicit goal of the InBlock design is to prevent the so-called jurisdictional overflow, where the allocations of entities in one country are under the jurisdiction of another country. We can phrase this in terms of fairness, i.e., that every entity has the right that its address allocations are not ruled by the legal framework of another country. This is not the case today, since allocations to entities based in the countries where the RIRs are based are ruled by their national law system, while for allocation to all other entities are ruled by foreign legal systems.

In particular, we identified the capability of entities placed in higher levels of the RPKI hierarchy of revoking existing allocations as a major concern springing from the jurisdictional overflow problem (e.g., a court of law of the country where a RIR is based can revoke an allocation obtained by an entity based in another country served by the RIR). With the InBlock approach, this problem no longer exists because, given the immutability of the underlying blockchain technology, there is no single entity capable of revoking an allocation. In other words, blockchain technology enables the creation of self-imposed restrictions in the management of a database. In the case of InBlock, one of such restrictions is that no entity can revoke an existent allocation. By giving up the revocation capability altogether, we provide certainty to prefix holders that no entity will be able to interfere with their address allocations.

F. Minimized overhead

By all accounts, the InBlock operation is very efficient and provides reduced overhead. As there are no humans involved in the operation, the costs are very low even considering the maintenance costs of the blockchain itself (about US$ 15, see section IV-G). Also, the time-scale of operation of the InBlock is significantly shorter than the current system. Allocations in InBlock are completed in the order of minutes.

G. Ethereum technological aspects

We now discuss technological aspects related to the specific platform of choice: Ethereum. We justify the selection of Ethereum for this purpose. Then we cursorily analyse if Ethereum can provide the latency, throughput and cost required by InBlock.

We design InBlock as a set of smart contracts on top of Ethereum. Previous proposals [38–40] propose to create a new blockchain to store information regarding IP address allocation (see section VII for further details about these proposals). In InBlock, instead of creating a new blockchain, we propose to use an existing one. We believe this approach provides three important benefits.

First, it provides a clean architecture with layered design that separates the blockchain from the registry service. This allows the evolution of the blockchain without affecting the registry service. For example, there is an ongoing debate regarding whether Proof of Work approaches are sustainable (due to their expensive cost in terms of power consumption) and whether Proof of Stake provides a more sustainable

alternative. By laying the InBlock on top of an existing (and evolving) blockchain, we make InBlock agnostic to the consensus mechanism. In particular, Ethereum currently uses PoW (which is the proven technology) and it is experimenting with PoS [41]. Once/if PoS is proven and stable, Ethereum will migrate to PoS and InBlock will benefit from this technological advance without any impact in the management of addresses.

Second, by using an existing blockchain, we can rapidly develop and deploy InBlock. For example, Ethereum is already available and working. By using Ethereum, we reduce the development time, as we only need to focus in the implementation of the registry service. All blockchain code evolution and testing[6] are taken care of by the Ethereum community.

Third, using an existing blockchain provides secure bootstrapping, i.e., a secure blockchain from the start of InBlock. Blockchain security heavily depends on the consensus mechanism used (e.g., PoW, PoS). The level of security of the consensus mechanism frequently depends on the level of adoption of the blockchain. A blockchain using PoW is as strong as the hashing power used to mine blocks [22]. In a new blockchain, it is likely that there will be little hashing power as there is little economical gain from mining it. If PoS is used instead, the level of security depends on the number of stakeholders, the concentration of stake and also how much actual value is there in the blockchain [22, 43]. In a new PoS blockchain, it is likely that there will significant concentration of stake and a reduced number of stakeholders control the blockchain. In the case of InBlock, the IPv6 addresses managed by the InBlock registry are valuable per se, prior to the existence of the blockchain. So, if the blockchain provides little security in its early days, there is a risk of an attack directed towards the illegitimate acquisition of allocations. The InBlock must then provide strong security from the bootstrap to be viable and using Ethereum achieves this goal.

Due to the facilities provided to develop smart contracts, and its relative maturity, Ethereum is our blockchain of choice. While the use of Ethereum seems to provide many advantages when building InBlock, we need to verify that Ethereum is also a good fit in terms of performance and cost. We next analyse different relevant parameters:

- **Timescale**. There are roughly three delays involved in an Ethereum transaction, namely, the time it takes for a miner to include a transaction in a block that it is mining, the time it takes to mine the block containing the transaction and the time it takes for the block containing the transaction to be confirmed. Regarding the first delay, miners receive many transactions that are candidates to be included in the next block to be mined. Miners determine which transactions may be included in the current block according to the transaction fee offered(transactions that are willing to pay more get in the blockchain earlier). According to current prices [44], if the InBlock transaction is willing to pay 2 US$ per allocation, the delay to be included in the next block is less than 2 minutes. Regarding the second type of delay, Ethereum publishes a new block every 17 seconds. Finally, assuming that InBlock will wait for 12 new blocks to confirm the transactions containing the allocations [45], this means that it will take about 3 more minutes to confirm the allocation transaction. So, the total delay for an InBlock transaction to be published and confirmed in Ethereum will be in the order of 5 minutes. Currently it takes days to grant new allocations, so the timescale provided by Ethereum is

[6]Notable bugs have been identified in different blockchains in the past [42].

much shorter than the one provided by the current allocation system.

- **Throughput**. Ethereum currently has a mean throughput of 20 transactions per second. In order to estimate the maximum throughput that could be required by InBlock, we compute the number of transaction that it would require renewing yearly all the current IPv4 and IPv6 allocations, plus the transaction resulting from the new IPv4 and IPv6 allocation done every year. This results in 58,700 transactions per year, which results in 0.0019 transactions per second, which is much less than the transaction throughput currently supported by Ethereum.
- **Cost**. In order to run in Ethereum, InBlock needs to pay to the Ethereum network in the form of a transaction fee. Executing InBlock implies the following expenses. First, the deployment of the InBlock code in the Ethereum network requires a transaction fee. We estimate roughly between US$ 15 and US$ 65 for this (depending on the gas cost selected). Once deployed, we estimate in around US$ 0.5 the costs of the transactions required to assign a block. There are other transactions required to run InBlock that will incur in additional costs (e.g., the cost of an oracle converting US$ to Ether is about 1 US$, setting a ROA costs around 0.1 US$). Note that the cost of a transaction fee is likely to increase as Ethereum becomes popular. If InBlock defines a fixed transaction fee according to current values, it is possible that this value will become outdated and that InBlock transaction become unattractive for miners to include them in new blocks they are mining. Because of this, the transaction fee used by the InBlock must be updated to reflect the values expected by the miners at every time. This can be done in Ethereum implementing an "oracle" [46] that provides the InBlock with updated values to use for transaction fees.

H. InBlock Deployment Considerations and Evolution

The deployment model we envision for InBlock is as follows. To operate, InBlock requires a block of globally unique IPv6 addresses to allocate. Once the block is assigned to the InBlock, it may prove to be hard to recover, so it is probably wise to be conservative and assign a small block (e.g., a /20). InBlock can then execute and perform allocations out of this initial block. If InBlock works as expected and all the addresses out of this initial block are allocated, it would be possible to launch other InBlock instances, each of them managing a separated address block. This deployment model allows to start with a small portion of the address space managed by the InBlock and grow if the demand increases, without jeopardizing the fate of a large address block from the start.

Different InBlock instances do not necessarily have to be equal, neither in size nor in encoded properties. Experience managing previous InBlock instances can inform modifications to be implemented in future ones. Moreover, it is also possible to envision different entities launching their own InBlock instances, e.g., if this service is proven to be valuable, different RIRs may decide to run their own InBlock instance. The deployment model of InBlock naturally supports this.

I. Limitations and open issues

Now we discuss some limitations of the proposed InBlock design that result from its automatic nature.

InBlock lacks the strong traceability features available in the IANA-RIR system for lawful purposes. Contact and other information associated to the allocations registered by the InBlock are only performed on

a purely voluntarily basis. This implies that misbehaving entities are unlikely to include any information that would be useful for their traceability. Networks participating in different types of attacks (denial of service, impersonation, etc.), should be identified through other means. We argue that this is an appropriate approach and that the traceability for legal reasons should be pursued through the ISPs providing connectivity to the allocated address block. The reason why we find this appropriate is that the InBlock provides a global service hence it should avoid the asymmetry existing in the current IANA-RIR system between the country hosting the registry and the rest of the countries. Pursuing legal actions through the ISPs providing connectivity to the allocated prefix seems to result in a better match between the scope of the organization and the scope of the legal framework.

One of the well-known shortcomings of cryptocurrencies is the dependency on the access to the private key associated to the account holding the tokens. In the case of InBlock, we have addressed this problem by requiring yearly renewal of the address allocation. In case the private key is lost, the address block would not be renewed, and the addresses will be returned to the pool. However, this will impose for the network previously using these addresses to acquire a new address block, and renumber its network, which is known to be expensive [47]. Moreover, if the private key is stolen (instead of lost) the robber can start using the addresses as his own (e.g., generating a new ROA), and there is no higher entity that the victim can appeal to. Note that this is a fundamental property of a system in which the ownership proof is tied to the control of a private key.

InBlock design brings predictability to address requesters, but it restricts the capability to react to unforeseen situations. For example, InBlock ties the evolution of the address allocation fee to the GDP. If due to some unexpected event there is a need to increase the fees this would be possible for new instances of the InBlock but not for existing ones. On the other hand, if the cost of the fees decrease, networks with addresses obtained through InBlock would face the dilemma of acquiring a new address block and experience the renumbering costs, or keep paying the higher fee derived by InBlock for the same resource.

V. InBlock Operation

In this section we describe the operation of the whole InBlock ecosystem. We start by describing the different roles of the actors involved in the IP address management process and the operations associated with them. Then, we briefly describe our implementation of the InBlock smart contract. Finally, we take a walkthrough that illustrates summarily how InBlock operates.

A. InBlock roles

The following roles with their associated operations are defined for InBlock operation:

- **InBlock Manager**. The InBlock Manager launches the execution of the InBlock smart contract in the blockchain. In order to do so, the InBlock manager obtains an IPv6 address block for the exclusive use of InBlock allocation, and configures it in the InBlock smart contract. the InBlock Manager also defines the size of the prefixes to be assigned, and sets the initial fees. Besides, it manages the Ethereum account that is used to receive the payments (i.e., it is able to transfer the funds received through the InBlock account and use them as needed).

- **InBlock LIR**. An InBlock LIR is any entity that requests a block directly from InBlock. It is entitled to renew its assignments when they are about to expire and can request additional address space contiguous to previously owned allocations, if this address space is available. It can use InBlock to publish contact and routing information associated to the blocks it owns, in addition (or alternatively) to the information currently being held by Internet Routing Registries (IRRs). Besides, InBlock LIRs can register in InBlock the transference of sub-prefixes to users, as this is an operation supported by the current address management system.
- **InBlock End User**. Borrowing the terminology of the RIRs for the entities which ultimately make use of the addresses they obtained from a LIR, we refer to the entities which have received a prefix delegation from InBlock LIRs as Inblock End Users. InBlock End Users are authorized to update the InBlock registries associated with their address allocation with contact and routing information. Part of the routing information that can be managed is the Route Origin Authorization (ROA), which specifies the AS identifiers which are authorized to generate a routing advertisement for the prefix.
- **InBlock Third Parties**. Any other entity can act as an InBlock Third Party by accessing freely to the InBlock information stored in the blockchain to identify if an address block has been assigned, and to the contact and routing information. In this way, anyone can use the InBlock chain-of-ownership to access, for example, the ROA of a prefix to perform route origin validation.

B. InBlock implementation

The InBlock smart contracts is code written in Solidity[48], a contract-oriented, high-level language for implementing smart contracts and designed to target the Ethereum Virtual Machine. Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features. An Ethereum smart contract is stored in the blockchain and is identified by means of an Ethereum account.

The smart contract includes *functions*, Solidity code which may change the contract state, e.g., the assignment of an address block. The modified state is stored in the blockchain as Ethereum transactions, and once registered in the blockchain, a node validating the blockchain will follow the smart contract's logic to determine its validity. Note that every node validating the blockchain checks the contract, not only nodes concerned with InBlock operation. On the other hand, a *call* consists of Solidity code that performs operations over blockchain data, but does not result in state changes. Therefore, it does not consume Ether, and is executed in the local system of the caller.

C. Walkthrough

To start the operation, the InBlock Manager configures the InBlock code with the relevant parameters for InBlock operation including the address block to assign, the length of the prefixes that will be allocated, and the fees (see Figure 4). Then, it transfers the code is to the Ethereum blockchain by doing a transaction. At this point, the InBlock is running and operational, ready to perform allocations.

We now consider a couple of cases to illustrate InBlock operation: Let's consider the case that an entity wants to become an InBlock LIR and receive the allocation of an address block. To do so, the entity creates an Ethereum blockchain account. Then, it accesses to the Ethereum blockchain, ensures

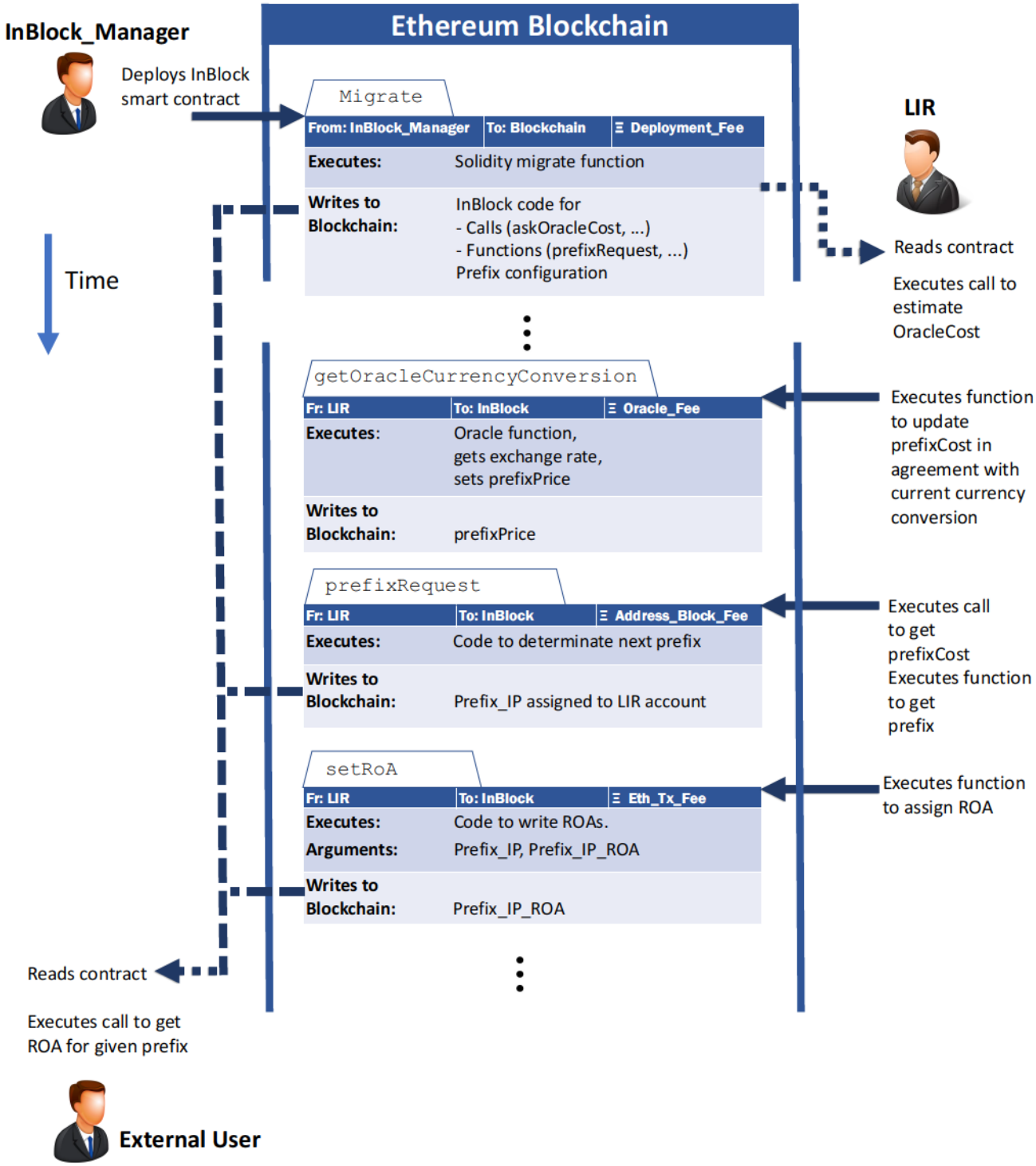

Fig. 4: Example of InBlock operation.

the information contained is valid according to Ethereum rules, and reads the InBlock contract. Before executing the code of the InBlock contract, the entity first needs to check that the contract has been deployed by the organization entitled to start the experiment, with the account that included the contract in the blockchain. The trust in the account authorized to run the experiment is similar to the trust placed in

the RPKI Trust Anchors. Once checked, the functions and calls contained in the contract can be executed by the LIR.

As discussed before, the block request operation relies on the payment of a fee. To determine the current value of the fee, stated as a fiat currency value, the caller needs to ask InBlock, which in turn uses an external service, an *oracle*, to compute the current value of the fee in Ether. This is done in a two-step process, in which the LIR-to-be executes a call (no transaction required) to estimate the fee of the oracle and then a function (a transaction) by which the InBlock code, with information from the oracle, writes into the blockchain the fee requested for the address block assignment. Encoding in the InBlock code the dependency on such a service poses two risks: that the service stops working (for example, because the infrastructure accessing to the physical world data feed stops working) or that the oracle returns false information (either intentionally or unintentionally). To provide protection against these risks, we can use many (for example, three) different oracles providing currency conversion and remove outlier values. The conversion value is obtained from the median of the three or, if one is not accessible, the mean of two. If less than two oracles are returned, a default value (set when InBlock is deployed) is used. In this way, we minimize the risk of all the oracles being offline and prevent the effects of a compromised oracle.

Then, the LIR executes the function to get the address block. The transaction attempt triggers the InBlock code that looks for free IPv6 blocks, selects one according to its rules, and stores as additional information to the transaction. As a result, an IP address block is assigned to the blockchain account of the requester. Now, the LIR can issue an Ethereum transaction to include a ROA in the blockchain. The InBlock code ensures that only the account associated with the legitimate owner of the prefix (according to the InBlock operation) can update the ROAs in the blockchain.

Consider next the case of an external user, a third party, which requires ROA information, for example, to prevent address spoofing by means of a route filter. RFC 8210 [49] describes `rpki-rtr`, an architecture in which routers offload to a trusted cache server the download of prefix origin data from the RPKI and its validation. Router and server communicate through a secured channel to keep their state synchronized. The integration of InBlock into the architecture defined by RFC 8210 requires the modification of the software at the `rpki-rtr` server to retrieve, validate and integrate InBlock's information into the prefix-to-origin AS state conveyed to the routers. To do so, the server accesses to the blockchain information, either as a full Ethereum node, and by other means. With the account identifier used to deploy the InBlock as trust anchor, the server retrieves the last valid ROA information for each prefix validly registered. Note that this operation is made through calls, so it does not involve any blockchain transaction (and the associated costs).

VI. Proposed experiment

As the blockchain technology is fairly new, constantly evolving and probably susceptible to bugs, and the IP address allocation apparatus is a cornerstone of the Internet, we are not proposing to create an operational IPv6 registry using InBlock. Instead, we are promoting an experiment with a limited scope, to gain a better understanding of how blockchain technology can be used to provide registry services for Internet resources and moreover, if the notion of a distributed autonomous registry is feasible.

Through the proposed experiment, we believe we can gain a better understanding of the following

aspects:

- Whether the proposed approached based on both a yearly fee and the abundance argument is enough to deter stockpiling.
- Whether the addresses obtained from the InBlock are actually used and announced in the Internet.
- Whether the blocks are announced as they are assigned, or extra deaggregation is observed.
- Whether a secondary market of addresses emerges.
- Whether the contact information of the allocated blocks is completed and updated with sound contents.
- How changes in Ethereum affect the InBlock service. In particular, when bugs appear, how much they affect the InBlock service and whether it is possible to cope with them. Also, how technological leaps in Ethereum affect the InBlock service (e.g., if Ethereum moves from PoW to PoS, how this affects InBlock).

The experiment, by definition, has a limited lifetime. The lifetime should be defined in advance, before starting the experiment. A reasonable time frame could be five years. If the experiment is a success, the lifetime of the experiment can be extended and the allocations made renewable for as long as the holders want to renew them. If the experiment is terminated, the allocations made through the InBlock should be transferred to one of the existent Internet registries upon its termination. This would allow organizations participating in the InBlock experiment to avoid the cost of renumbering regardless the result of the experiment. With this conditions, the InBlock experiment is likely to be more realistic, as more organizations may use the resources obtained through the InBlock in real operations.

The experiment should include some safeguards in case an unexpected behavior is observed. For instance, it would be beneficial to rate limit the number of allocations made. If the rate in the applications is excessive, the experiments should be paused and the control passed to human supervisors.

In order to run the InBlock experiment, a pool of globally unique addresses is needed. The organizations that can provide the required pool of addresses and run the experiment are:

- The IETF (80% of the IPv6 address space is “Reserved for the IETF” [50], and it is up to the IETF to instruct IANA to allocate an address block for the InBlock experiment),
- ICANN, through the IANA,
- The RIRs, either one of them on its own, or jointly through the NRO
- A NIR (National Internet Registry).

In addition of providing the addresses required to perform the experiment, the organization running the experiment should also be capable of properly designing and implementing it. In this paper we proposed InBlock that could serve as a starting point for that debate, but in order for the experiment to be useful, the Internet community should be actively engaged in the ultimate design of the experiment. Similarly, once the experiment has been defined, it needs to be implemented and tested. The resulting implementation should be reviewed by the community to make sure that there are no errors or backdoors. We believe that there are several organizations who have the capabilities to run the proposed experiment, such as the IETF or the RIRs. Moreover, if the experiment is proven to be successful, the IETF and/or the RIRs could naturally have a stake in the distributed autonomous organization.

VII. Related work

We next briefly describe existing work related to management of Internet resources using blockchain technology. In particular, we attempt to identify how the previous work has addressed the challenges pointed out in section IV.

Namecoin [40] is a working system that provides decentralized namespaces improving security, privacy and censorship resistance.

Namecoin supports name resolution for several namespaces, including the DNS names within the `.bit` domain. While the `.bit` domains belong to the globally DNS hierarchy, the mechanism for resolving `.bit` names is different than for other DNS names, as it requires querying the Namecoin blockchain for name resolution. This implies changes in the host resolver. Technologically, Namecoin is a fork of Bitcoin with a new native token, the Namecoin (NMC). Namecoin then uses a PoW consensus mechanism. The main challenge when using PoW for a new coin is the lack of economic incentives for miners to mine the new coin. In order to address this issue, Namecoin uses merged-mining with Bitcoin. Namecoin DNS names are paid using NMC tokens. The price of the NMC token is defined by the market. In theory, the NMC token price would be high enough to prevent stockpiling. In reality, there was an initial peak on the demand for attractive names such as well-known trademarks. Analysis showed that most of these names belong to three entities, and are not used for name resolution, so it should be assumed that were acquired for speculation. After this initial phase, the price of the NMC token drop to its current value that is close to US$ 0.

As a take away, Namecoin is a valuable experience for designing blockchain based systems for managing Internet resources. First, the Namecoin experience shows that the fee mechanism must use a high-enough fee to be effective. Leaving the market to determine the fee seems unwise, especially in the early phase, when there is little demand, the market laws make the price to decrease, which in turn render the fee mechanism ineffective.

However, there are fundamental differences between IP addresses and DNS names, namely: First, while all IP addresses have the same value, DNS names do not. Some DNS names are more attractive than other (e.g., those reflecting trademarks). Second, in order to use `.bit` names allocated though Namecoin, host modifications are required, while no modification in end hosts are required to use IP addresses allocated through the InBlock.

Internet Blockchain [38] propose the use of blockchains to register IP addresses. They propose the design of a Bitcoin-like blockchain for the management of all Internet resources including IP addresses, ASes, DNS and BGP route. The main goal is to avoid centralization and the control of a single authority. To reduce the design and development risks associated to the blockchain technology, they choose the Bitcoin as its underlying platform. They propose using *multi-sig*, a technology in which a cryptographic operation can be performed or validated with a minimum number of keys from a larger set, to alleviate the key-loss problem. They conclude that this technology is still not mature enough to support path validation, i.e., path authenticaiton for BGP route advertisement. For this reason, they propose an incremental deployment, starting from the management of RPKI functionalities, then supporting route advertisements, and finally providing name resolution (currently solved by the DNS system). Although they consider, as we do, the idea of replacing the source origin validation provided by RPKI with a blockchain-based mechanism,

they do not address the design of an automatic address assignment system (with key concerns such as stockpiling prevention or secure bootstrapping).

Palisse et. al. [39] propose a blockchain mechanism to manage IP addresses. They propose to build up a new blockchain to record IP addresses allocation and delegation, based on a PoS consensus mechanism, where addresses are the main asset. The main concern regarding such approach is that PoS results in major stakeholders having more chances to mine new blocks. This means that the current authority of the global IPv6 address pool (IANA) would certainly control the majority of the stakes and probably the blockchain, defeating the goal of preventing centralized control of the IP address allocation system.

Like Namecoin, they propose an expiration mechanism with renewal to prevent the lost-keys problem and to preserve the consistency of the Internet resources. This work does not consider concerns such as stockpiling or secure bootstrapping.

Kuerbis and Mueller [6] discuss the applicability of blockchain technology to perform the Internet Routing Registry functions. Unforgeability and a provable timeline are identified as desirable properties of a routing policy registry, and are naturally provided by blockchains. They consider a system in which operators could cryptographically point to routing policy information repositories by using Blockstack, a blockchain to facilitate decentralized naming systems. Integration with RPKI key data can ensure the authenticity and integrity of the data. InBlock is similar in the sense that it stores the minimal metadata in the blockchain, and points elsewhere to the storage of the information. In this case, the authenticity and integrity of the routing policy data is the cryptographic link to the identity to which the addresses where allocated.

A. Alternative IP address allocation systems

There have been proposals for alternative mechanisms to manage IP address allocations. In particular, there have been several proposals for geographically aggregatable addresses (the original IPv6 address architecture reserved an address range for geographic-based unicast addresses [51]). The main concern with geographic aggregation is that it opposes to the business logic of the ISPs, since it would require ISPs covering a given geographic area to announce the routes for the prefixes of that area and thus carry traffic for customers of the competing ISPs in the region. InBlock does not argue for any form of geographical aggregation, and thus, suits to the current business model. Addresses allocated by InBlock can be used as any other Global Unicast Address assigned through the current IANA-RIR based system.

Transferable Block Lease (TABL) [52] proposes a market-oriented IPv6 allocation mechanism. They suggest that RIRs set aside an address block that can be allocated without the assessment of the needs to the applicant, with a fee that is related to the size of allocation. The block allocated though TABL would be transferable between parties. They argue that such an approach would provide a simpler method for address management that would also reduce the provider lock-in problem, as end users would be able to have direct access to addresses that they could keep after a provider switch. Address conservation is enforced by the economic incentives for resource holders to request the amount needed, and trade or return the unused address blocks. They also suggest performing an experiment to assess the benefits and risks of the proposal.

The address assignment mechanism described by TABL is similar to InBlock in the sense that they

both rely on fees to access to IP address allocations without an assessment of the address needs of the applicant. They are also related in the analysis of how the resulting mechanism would conserve the address space. However, TABL and InBlock differ in their main objectives: InBlock aims to prevent jurisdictional overflow, while TABL intends to facilitate to ISPs and end users access to address blocks. They also differ in their nature, as TABL is an address assignment policy implemented through contractual framework which still relies in the IANA-RIR for the whole address management process. Consequently, the same entities of the current system (RIRs, NIRs) retain full control for registering and issuing the certificates required for RPKI operation. As IANA and the RIRs still control the address management, they can prevent parties to obtain addresses, revoke leases and invalidate its associated RPKI information.

Some of the mechanisms proposed by TABL could be considered for InBlock adoption. Support for address transferability can be implemented as an InBlock function enabling the modification of the account that manages an address allocation. TABL advocates the support for a wider range of address block sizes, any size between /32 and /48. In this case, a large address space block should be reserved for InBlock to ensure for each of the block sizes that a sparse allocation strategy can be used to enable an entity to request a contiguous block to a previously allocated one.

VIII. Conclusions and final remarks

In this paper we have presented the design and implementation of InBlock, a decentralized autonomous organization that is capable of performing the IPv6 global registry functions without human intervention. We have designed the mechanism to create the incentives so that the system can autonomously prevent different form of abuses, including stock-pilling and other wasteful techniques, and thus, preserve the address space. Because the InBlock is decentralized, as its execution is performed by the different miners all around the planet, the InBlock also addresses the identified jurisdictional overflow problem.

The clear and transparent rules defined may appeal to organizations facing legal uncertainty costs due to their exposition to multiple legal jurisdictions. Moreover, InBlock's prefix allocation incurs in very low operational cost, tens of US$ (that is increased to serve address conservation purposes) and very low delay, tens of minutes, as discussed in section IV-G.

There are many different ways that blockchains can be used to build a global IP address registry. In particular, InBlock implements a registry that is not under control of any single entity, implying that no single entity can prevent another entity to obtain IP address allocations. InBlock provides quite strong privacy guarantees and censorship resistance. Nevertheless, as a trade-off, InBlock gives up the traceability and enforcement features. This means that the IPv6 address registry functions provided by InBlock cannot be used to trace the identity of the holders of the resources if they do not voluntarily include contact information in the registry. Similarly, the InBlock cannot be used to restrict the Internet access to any entity by revoking their address allocations (actually it is a goal of the InBlock to prevent this situation).

The InBlock design described here aims to achieve some defined goals (uniqueness, stockpiling prevention, aggregation preservation, etc.) by means of some mechanism (blockchain consensus, pricing, limits on the number of AS number involved in the delegations, etc.) The extent to which these goals can be achieved with these measures can only be derived from real experience. We also appreciate that there is not enough experience in the use of blockchains for solving real-world problems. Because of this,

this is a call to arms for experiments on the use of blockchain technology for implementing Distributed Autonomous Organizations in general and for implementing IP registry functions in particular. We propose to experiment with InBlock for IPv6 address management. We believe this experiment will provide useful hand-on experience about blockchain-based organizations.

ACKNOWLEDGEMENTS

This work was supported by Huawei through the InBlock HIRP prize and by the Spanish Ministry of Economy and Competitiveness through the 5G-City project (TEC2016-76795-C6-3-R).

REFERENCES

[1] H. Zhao, "ITU and Internet Governance," *ITU document WG-WSIS-7/6 Rev 1*, 2004.

[2] L. R. E. Heilman, D. Cooper and S. Goldberg, "From the Consent of the Routed: Improving the Transparency of the RPKI," *Proceedings of the 2014 ACM conference on SIGCOMM*, Aug 2014.

[3] V. Buterin, "The Meaning of Decentralization." https://medium.com/@VitalikButerin/the-meaning-of-decentralization-a0c92b76a274, Feb 2017.

[4] L. Levin and Stephen Schmidt, "IP4 to IPv6: Challenges, solutions and lessons," *Telecommunications Policy, vol. 38*, 2014.

[5] K. Hubbard, M. Kosters, D. Conrad, D. Karrenberg and J. Postel, "Internet Registry IP Allocation Guidelines." IETF RFC 2050, 1996.

[6] B. Kuerbis and M. Mueller, "Internet routing registries, data governance and security," *Journal of Cyber Policy*, 2017.

[7] R. Housley, J. Curran, G. Huston and D. Conrad, "The Internet Numbers Registry System." IETF RFC 7020, Aug. 2013.

[8] M. Mueller, "Critical resource: An institutional economics of the Internet addressing-routing space," *Telecommunications Policy*, 2010.

[9] ICANN, "Stewardship of IANA Functions Transitions to Global Internet Community as Contract with U.S. Government Ends." https://www.icann.org/news/announcement-2016-10-01-en, Jul 2018.

[10] ICANN, "Stewardship of IANA Functions Transitions to Global Internet Community as Contract with U.S. Government Ends." https://www.icann.org/resources/pages/governance/litigation-en, May 2018.

[11] RIPE, "IPv6 Address Allocation and Assignment Policy." https://www.ripe.net/publications/docs/ripe-699\#minimum_allocation, May 2018.

[12] APNIC, "IPv6 Address Allocation and Assignment Policy." https://www.apnic.net/community/policy/ipv6-address-policy_obsolete/, May 2018.

[13] ARIN, "IPv6 Address Allocation and Assignment Policy." https://www.arin.net/vault/policy/archive/ipv6_policy.html, May 2018.

[14] LACNIC, "IPv6 Address Allocation and Assignment Policy." http://www.lacnic.net/684/2/lacnic/4-ipv6-address-allocation-and-assignment-policies, May 2018.

[15] AFRINIC, "IPv6 Address Allocation and Assignment Policy." https://afrinic.net/fr/library/policies/122-afpub-2004-v6-001, May 2018.

[16] RIPE, "What are Provider Aggregatable (PA) addresses and Provider Independent (PI) addresses?." https://www.ripe.net/participate/member-support/copy_of_faqs/isp-related-questions/pa-pi, May 2018.

[17] B. Carpenter and S. Jiang, "Significance of IPv6 Interface Identifiers." IETF RFC 7136, Feb. 2014.

[18] M. Lepinski and S. Kent, "An Infrastructure to Support Secure Internet Routing." IETF RFC 1884, Feb 1995.

[19] NIST, "RPKI Monitor NIST." https://rpki-monitor.antd.nist.gov/, Dec 2018.

[20] Number Resource Organization, "Regional Internet Registries are preparing to deploy "All Resources" RPKI Service." https://www.nro.net/regional-internet-registries-are-preparing-to-deploy-all-resources-rpki-service/, May 2018.

[21] Z. Zheng, S. Xie, H. Dai, X. Chen and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. IEEE International," *BigData Congress*, 2017.

[22] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei and C. Qijun, "A review on consensus algorithm of blockchain," *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017.

[23] M. E. Peck, "How Smart Contracts Work," *IEEE, Spectrum*, 2017-09.

[24] L. Yu, G. Li, Y. Yao, C. Hu and W. Tsai and E. Deng, "Smart-Contract Execution with Concurrent Block Building," *IEEE Symposium on Service-Oriented System Engineering*, 2017.

[25] V. Buterin, "Ethereum White Paper." https://github.com/ethereum/wiki/wiki/White-Paper, May 2018.

[26] V. Buterin, "DAOs, DACs, DAs and More: An Incomplete Terminology Guide." https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide/, May 2018.

[27] S. DAVIDSON, P. DE FILIPPI, and J. POTTS, "Blockchains and the economic institutions of capitalism," *Journal of Institutional Economics*, vol. 14, no. 4, p. 639–658, 2018.

[28] C. Team, "CoinMarketCap." https://coinmarketcap.com/all/views/all/, May 2019.

[29] J. Desjardins, "All of the World's Money and Markets in One Visualization." http://money.visualcapitalist.com/worlds-money-markets-one-visualization-2017/, October 2017.

[30] K. Fanning and D. P. Centers, "Blockchain and its coming impact on financial services," *Journal of Corporate Accounting & Finance*, vol. 27, no. 5, pp. 53–57, 2016.

[31] A. Savelyev, "Copyright in the blockchain era: Promises and challenges," *Computer Law & Security Review*, vol. 34, 12 2017.

[32] D. Yermack, "Corporate Governance and Blockchains*," *Review of Finance*, vol. 21, pp. 7–31, 01 2017.

[33] P. De Filippi and S. Hassan, "Blockchain technology as a regulatory technology: From code is law to law is code," *First Monday*, vol. 21, 11 2016.

[34] A. Durand and C. Huitema, "The Host-Density Ratio for Address Assignment Efficiency: An update on the H ratio." IETF RFC 3194, 2001.

[35] WorldBank, "GDP, current US$." https://data.worldbank.org/indicator/NY.GDP.MKTP.CD, Dec 2018.

[36] J. Roberts and N. Rapp. Buterin and V. Griffith, "Exclusive: Nearly 4 Million Bitcoins Lost Forever, New Study Says. Fortune." http://fortune.com/2017/11/25/lost-bitcoins/, Jun 2018.

[37] P. Wilson, R. Plzak and A. Pawli, "IPv6 Address Space Management," *RIPE-343*, Jun 2018.

[38] A. Hari and T.V. Lakshman, "The Internet Blockchain: A Distributed, Tamper-Resistant Transaction Framework for the Internet," *HotNets-XV*, 2016.

[39] J. Palisse, A. Rodriguez-Natal, V. Ermagan and A. Cabellos, "An analysis of the applicability of blockchain to secure IP addresses allocation, delegation and bindings," *IETF draft-paillisse-sidrops-blockchain-01, work in progress*, 2017-10.

[40] H. Kalodner, M. Carlsten, P. Ellenbogen, J. Bonneau and A. Narayanan, "An empirical study of Namecoin and lessons for decentralized namespace design," *Workshop on the Economics of Information Security (WEIS)*, Jun 2015.

[41] V. Buterin and V. Griffith, "Casper the Friendly Finality Gadget." https://vitalik.ca/files/casper_note.html, May 2018.

[42] M. Suiche, "The $280M Ethereum's Parity bug." https://blog.comae.io/the-280m-ethereums-bug-f28e5de43513, May 2018.

[43] "Proof-Of-Stake." https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQ, May 2018.

[44] "ETH Gas Station." https://ethgasstation.info/, May 2018.

[45] V. Buterin, "On slow and Fast Block Times." https://blog.ethereum.org/2015/09/14/on-slow-and-fast-block-times/, May 2018.

[46] "Gas Price Oracle." https://github.com/ethereum/go-ethereum/wiki/Gas-Price-Oracle, May 2018.

[47] B. Carpenter, R. Atkinson and H. Flinck, "Renumbering Still Needs Work." IETF RFC 5887, 2010.

[48] E. Team, "Solidity Doc." https://solidity.readthedocs.io/en/v0.4.25/, Dec 2018.

[49] R. Bush and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 1." IETF RFC 8210, 2017.

[50] "IPv6 Address Space." https://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xhtml, May 2018.

[51] R. Hinden and S. Deering, "IP Version 6 Addressing Architecture." IETF RFC 1884, 1995.

[52] M. Mueller and Y. Kim, "Economic Factors in the allocation of IP addresses," *International Telecommunication Union*, 2009.

Stefano Angieri received a computer science bachelor degree in 2014 and a master degree in 2018 at Università degli studi Federico II di Napoli with a master thesis on blockchain technology. In 2018 he joined Universidad Carlos III de Madrid (UC3M) as Ph.D student. His main interest area is blockchain technology.

Alberto García-Martínez received a telecommunication engineering degree in 1995 and a Ph.D. in telecommunications in 1999. In 1998 he joined Universidad Carlos III of Madrid (UC3M), where he has been an associate professor since 2001. His main interest areas are interdomain routing, transport protocols, network security and blockchain technology. He has published more than 50 papers in technical journals, magazines, and conferences, and has also co-authored three RFCs.

Bingyang Liu received his Ph.D. degree from Department of Computer Science, Tsinghua University in 2014, and he was a joint Ph.D. student in Duke University from 2011 and 2012. Since 2014, he worked as postdoctoral research associate in Institute for Network Sciences and Cyberspace, Tsinghua University. He joined Huawei Network Technology Lab in 2016, and he is now a principal researcher. His research interests include network architecture, network security and trustworthiness, decentralized network infrastructure, routing and name resolution, congestion control, deterministic network, and future Internet. He is active in academia and industrial SDOs, including IETF, ITU-T, ETSI and CCSA.

Zhiwei Yan received his Ph.D. degree from National Engineering Laboratory for Next Generation Internet Interconnection Devices at Beijing Jiaotong University. He joined China Internet Network Information Center in 2011 and is currently a full-time Professor. Since April 2013, he has been an Invited Researcher of Waseda University. He is active in APNIC, ICANN and IETF. He published RFC 8191 in 2018. His research interests include mobility management, network security, and next generation Internet.

Chuang Wan joined Huawei in 1998. He is now a senior technique expert and architecture of Huawei Network Technology Lab. He is a senior expert in network technologies, protocols and product solutions. He is responsible for exploratory research on future network architecture and protocol evolution technologies. His research filed covers next-generation protocol architecture evolution, future core router architecture innovation, self-organization network protocol and architecture, and data center network solutions. He was a system architecture of Huawei NE series routers, chief planning expert of Huawei network operating system, and chief planning expert of Huawei IP-area products.

Marcelo Bagnulo (h-index=23, total citations=3631) received the Electrical Engineering degree and the Ph.D. in Telecommunications in 2005, from Universidad Carlos III de Madrid (UC3M), Spain. He holds a tenured associate professor position at UC3M since 2008. His research interests include Internet architecture and protocols, inter-domain routing and security. He has published more than 60 papers in the field of advanced communications in journals and congresses (including INFOCOM and IEEE/ACM Transactions on Networking) and he is the author of 18 RFCs in the Internet Engineering Task Force (IETF) including the Shim6 protocol for IPv6 multihoming and the NAT64/DNS64 tools suite for IPv6 transition. Dr. Bagnulo was a member of the Internet Architecture Board between 2009 and 2011.