BALAdIN: truthfulness in collaborative access networks with distributed ledgers

Vincent Messié, Gaël Fromentoux, Nathalie Labidurie, Benoit Radier, Sandrine Vaton, Isabel Amigo

Annals of Telecommunications manuscript No.
(will be inserted by the editor)

BALAdIN: Truthfulness in collaborative access networks with Distributed Ledgers

Vincent Messié · Gaël Fromentoux · Nathalie Labidurie · Benoit Radier · Sandrine Vaton · Isabel Amigo

Received: date / Accepted: date

Abstract Distributed Ledger Technology (DLT), also known as “Blockchain” is one of the trendiest digital innovations in our era. While firstly applied to cryptocurrencies, the trustworthiness inherent to DLT paved the way to many new usages notably in sectors such as land registration or banking where confidence in transactions is crucial. In the telecommunications sector, DLTs have enabled the development of future network architectures, such as new decentralised and multi-actor access networks. Having identified this as a great opportunity, we designed BALAdIN, a novel Blockchain-powered decentralised application that improves network coverage thanks to partnerships. Indeed, it allows a crowd of local actors such as shop tenants to deploy mobile cells with the help of a consortium of telcos. The traffic conveyed by each actor is traced thanks to a decentralised and trustworthy network monitoring mechanism we have designed. This mechanism both solves the centralisation dilemma caused by Blockchain Oracles and allows each actor to be rewarded depending on usage.

In this paper, we focus on the description of this network usage monitoring mechanism. We then study the feasibility of its implementation onto regular Blockchain by studying its performance regarding the throughput and the propagation of the resulting Blockchain transactions. Finally, we derive a scalable deployment scheme for our novel Blockchain-powered decentralised network metering application BALAdIN.

Keywords Blockchain · Distributed Ledger Technology · DApp · collaborative network · last mile network

Vincent Messié, Gael Fromentoux, Nathalie Labidurie and Benoit Radier
Orange Labs
2 Avenue Pierre Marzin
22300 Lannion France
E-mail: firstname.lastname@orange.com

Vincent Messié, Sandrine Vaton and Isabel Amigo
IMT Atlantique
655 Avenue du Technopole
29280 Plouzané France
E-mail: firstname.lastname@imt-atlantique.fr

1 Background

Distributed Ledger Technology (DLT), which rose to fame in 2008 with the Bitcoin currency [1], is the founding technique of cryptocurrencies. Indeed, this distributed database secured with cryptographic variables provides inherent trust. Furthermore, it features embedded consensus mechanisms such as Proof of Work (PoW) or Proof of Stake (PoS) [2]. A DLT ledger is then self-regulated and secured by a network of "validator nodes" cooperating to perform the consensus mechanism. Because the trust it provides to all users is inherent, many decentralised applications have adopted DLT.

The most commonly used distributed ledger structure relies on a single chain of blocks, hence its name is known as "Blockchain". Indeed, transactions are packed into blocks, and these blocks are chained with each other, one after the other. Blockchain thus refers to a specific DLT implementation. These ledgers can be seen as a huge book that anybody can read, to which anyone can add new pages, provided the page is approved by other users, but where it is impossible to alter existing data.

While Distributed Ledgers are mainly used to power decentralised currencies, new use-cases are emerging thanks to "smart-contracts", which appeared with Ethereum [3]. A smart-contract is a piece of software, deployed on all the Distributed Ledger nodes taking part in the validation process, and used to customise the validation process of a transaction. Thus it governs the creation of the transactions that are to be stored in the Distributed Ledger.

Applications using Distributed Ledgers and a set of smart-contracts are called "Decentralised Applications".

While many Distributed Ledgers are public and fully open, a "permissioned" private Distributed Ledger implementation is possible if the access to the ledger needs to be restricted. Alternative "hybrid" implementations are also possible. Some of their features are permissionless, while others are permissioned. Ripple [4] is an example of one such hybrid Distributed Ledger. From a user perspective, it is a public and permissionless Distributed Ledger: users can read the ledger but cannot add information. Its governance is based on a permissioned consensus, improving security and transaction throughput.

Permissionless Distributed Ledgers help the federation of multiple actors thanks to their native trust. They can for instance be used to track agreements between a consortium of telcos (sharing a common governance) and multiple partners, avoiding the need for a trusted third party such as a Data Clearing House.

As a consequence, many telecom-related projects are emerging. Helium [5] aims at building the "world's first decentralised wireless network", thanks to a dedicated Blockchain, and a unique consensus mechanism called a "Proof of Coverage". This mechanism encourages users to provide wireless network coverage, by rewarding them for the use of the resources they provide. Similarly, Ammbr [6] intends to build a decentralised mesh network thanks to a dedicated Blockchain, and dedicated hardware, where connectivity would be autonomously traded on a marketplace. PayFlow [7] aims at making a DLT-based solution to automatically trade network bandwidth thanks to micro-payments.

The Carrier Blockchain Study Group, CBSG [8] is another example of a global cross-carrier Blockchain platform. It seeks to provide a DLT solution to connect the carrier's telecommunication back-ends, to enhance transaction time and avoid failures between telecom operators.

In the Telco ecosystem, the TMForum is exploring new business and network models based on the DLT. Indeed, such models would allow distinct actors to cooperate and create enhanced connectivity services. For that purpose, a Catalyst has been set up to explore new federated network solutions, including its business and technical aspects [9].

Furthermore, the Technical Report TR279 [10] introduces the technology and its unique features, and then highlights various use-cases for service providers.

Torcoin is another interesting design imagined in 2014 [11]. This solution provides a way to reward actors for providing resources, known as TOR Onion Router (TOR) relays [12], in a decentralised and anonymous way. Indeed, TOR relays provide resources to TOR networks, by forwarding the traffic, either to another TOR relay or to the open Internet. They further encrypt the traffic and mask sensitive information such as IP addresses. As a result, every relay of the TOR network is kept anonymous, and therefore rewarding them is a challenge, despite their contribution to the network. Torcoin solves this issue by offering incentives thereby encouraging users to act as relays.

For that purpose, Torcoin uses a Blockchain with a novel consensus mechanism called a Proof of Bandwidth (PoB). Unlike Bitcoin, whose security is ensured by computational power, Torcoin coins are mined by the relays themselves, the ones providing network resources to the users. To achieve this, a specific frame is regularly sent along the TOR path, composed of a client, an inner relay, a middle relay and an outer relay. This frame performs a round trip (from the client to the outer relay, and back). During this round trip, users exchange cryptographic variables and digitally stamp the frame. As such stamps ensure the integrity of the frame, it carries the proof of the existence and the usage of the TOR path. This frame is then saved onto the Torcoin Blockchain by the client, which will reward the path relays, thanks to newly mined coins.

However some users may easily exchange their private keys to generate a fake PoB, hence creating a high risk of collusion [11]. This is why Torcoin relies on "assignment servers" to guarantee a certain level of security. The PoB paths are actually created by these trusted third parties, shuffling the user's matrix to reduce the risk of collusion. As all of the path's users need to collude to corrupt a PoB, only 1 out of 16 TOR paths would be corrupted if half of the users were colluding [11].

In [13] we introduced BALAdIN, a novel collaborative network architecture. On the one hand, BALAdIN relies on a consortium of telcos that share their infrastructure to maximise its usage. On the other hand, BALAdIN relies on a crowd deployment model, encouraging "local actors" such as shop tenants or restaurants to deploy cells and thus locally improve the overall network coverage. A customer of any telco involved in the Consortium could then transparently connect to the deployed small cell.

The BALAdIN network takes advantage of both DLT and Torcoin's PoB process to foster trust and collaboration between the stakeholders (customers, local actors and telcos). In BALAdIN the PoBs are used to certify the amount of network traffic exchanged. As a consequence, they provide valuable and trustworthy information that may be used for accounting and then billing purposes.

In this paper, we start by describing our solution in more detail. We then study the deployment of a DLT onto a distributed and multi-party infrastructure, giving special insight into its expected performance and taking into consideration the limited scalability of the Blockchain.

2 Motivations

With the constant demand for better coverage and more bandwidth, network operators have to engage in the transition to 5G while dreading the costs. Novel network architectures must be imagined to sustain the deployment of future networks.

In traditional models, telcos have managed and only used their own infrastructures, which has led them to reach their limits. To ensure a more sustainable development of networks, we believe that new paradigms must be considered. Why not then start by making telcos cooperate in order to share their existing infrastructures and thus maximise their potential? Such an operator consortium could also rely on crowd deployment with multiple local actors like shop tenants or restaurant owners to densify the existing coverage with the help of small cells. Moreover, such collaborative systems would allow the cooperating telcos to use an infrastructure they don't own. Yet confidence is key if such a system is to succeed. Proper agreements and policies are essential to ensure stakeholders cooperate in a dependable and secure way.

This paper addresses a hot topic as, now more than ever, trust between digital players such as connectivity providers is more than necessary. Indeed, the fifth-generation mobile networks will embed multiple innovations such as slicing, D2D communications, automation, etc. to accommodate future needs of a fully digitalized society [14].

Furthermore, Mobile Edge Computing or Multi-access Edge Computing (MEC) is soon going to spread, as putting computing resources at the edge has been identified as a requirement for sustaining the development of new applications. Yet such a leap will require the cooperation of telcos and Cloud resource providers. This challenge has even been identified as the hardest hurdle to overcome [15,16].

The cited contributions show that the necessary trust prerequisites can be complicated to meet as they require advanced features like Data Clearing Houses. On the other hand, using DLT to unleash the potential of decentralised Networks is now well-documented, for many projects are emerging as shown in [5–8].

Furthermore, TMForum's TR279 [10] specifically addresses the advantages of DLT for network service providers, by reviewing the key features of the technology, and proposing novel DLT-based use-cases. Specifically, Table 1 lists some DLT characteristics analysed in this report, and their relationship with various use-cases, including ours, thus showing the pertinence of the DLT for our use-case.

The true potential of DLT lies in its capacity to make multiple and distinct stakeholders trust each other. DLT creates almost-unbreakable bonds thanks to a shared unalterable ledger. This then allows new connectivity use cases to emerge, such as the cited contributions.

However, we think that for such decentralised applications, the data fed into smart-contracts must be trustworthy prior to its use. Thus, we avoid introducing a centralised Oracle in the Distributed Ledger. For a smart contract, an Oracle

Table 1 Some DLT use-cases: categories of interest as described in the Catalyst Technical Report TR279 [10]. Rows represent some DLT key features while columns represent use-cases using those features

DLT Features	Billing	Identity	Traffic monitoring (BALAdIN)
Works in a "trust-less" environment	X	X	X
ledger immutability	X	X	X
Easily auditable	X	X	X
Decentralised Currency	X	X	

is the interface to the outside world when external information such as network performance needs to be gathered. Indeed, as it is impossible to alter existing data on a Distributed Ledger, there must be a way to verify and secure the incoming data flow. Introducing such an entity is difficult to perform and may lead to a corruption/failure of a ledger [17], due to the possible introduction of a single point of failure.

Corruption is a big problem as smart-contracts are supposed to be foolproof. The attack on "*TheDAO*" organisation [18] is a good example of what happens when a smart-contract fails. In that specific case, this resulted in a "hard fork" of the Blockchain: some users decided to ignore the corrupt blocks and start the Blockchain over from a specific point. This illustrates the problem of introducing points of failure in a Blockchain application. Whereas the PoB mechanism introduced above provides a decentralised way to measure the traffic on a given network path using PoB frames [11]. The final frames could indeed be packed into "PoB Transactions" to be used in decentralised applications, as the whole PoB mechanism provides consensus-secured data that can be trusted by every player involved in the process. A decentralised and foolproof access network is therefore possible using Blockchain-powered smart-contracts, and the PoB protocol to feed them with consensus-secured data.

As a result, we propose a "hybrid" approach for BALADIN, with both a Distributed Ledger to store connectivity metrics thanks to PoBs, and a trusted consortium of operators for accounting, auditing, billing and possibly reputation. Indeed, whereas the Distributed Ledger Technology is a good compromise to power our solution in an efficient way whilst avoiding too much centralisation, a trusted consortium of operators shall enforce the global trust in our solution. Actually, the operators firstly perform the authentication of users and eventually manage a reputation system and by doing carry out the necessary audits on the ledger.

3 The BALAdIN solution in detail

In [13] we introduced BALAdIN, a collaborative solution powering 5G distributed networks. The solution allows a crowd of "local actors" such as shop tenants, railway stations, supermarkets, etc. to deploy small cells managed by their connectivity provider and get a reward based on the use of these cells. Users of such a system could then benefit from better coverage, thus improving their experiences at greatly reduced costs.

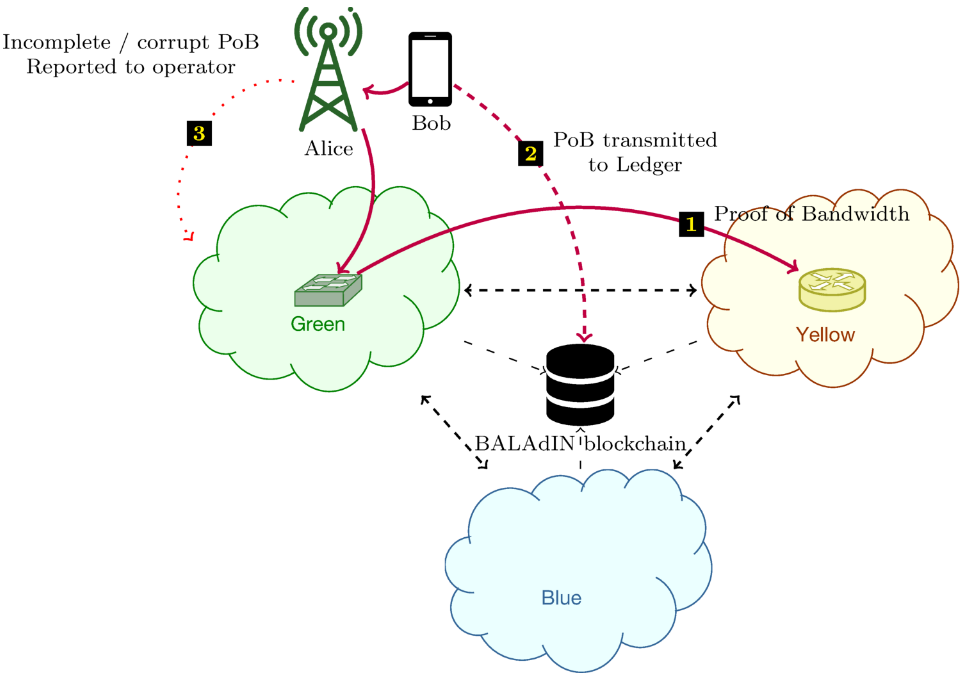

Fig. 1 The BALAdIN design. In this example, three operators form a consortium. Thanks to agreements between Green and Yellow operator, Bob, a customer of Yellow operator can get broadband connectivity from Alice, a local actor customer of Green operator. A hybrid Blockchain is set up between all actors to track network-related events in a decentralised way thanks to Proofs of Bandwidth. Yet this Blockchain remains permissioned as users and local actors are authenticated by their respective operator central authorities. In **1**, PoBs are exchanged within the path then stored and secured onto the ledger (step **2**), thanks to agents deployed on the network nodes. If the PoB process fails for whatever reason this should be reported to telcos (step **3**).

In the example presented in Figure 1, Alice, a local actor, has deployed a small cell with the help of Green, her operator. The deployed cell will grant Bob, a travelling customer of Yellow, some mobile connectivity to reach its home operator network.

From the business side, such a platform has already been proposed by promising projects [5–7]. We are thus focusing on the technical side of the solution.

Similarly, the telcos cooperation model won't be covered in this paper, at it is outside the scope of our work.

We thus focus our work on the Blockchain keeping track on the use of the network with the help of PoB stored on the Blockchain.

The PoB mechanism is then used to provide a decentralised way of measuring the traffic, with the help of agents deployed on strategic nodes along the path. Although the presence of telecom operators in the access network is no longer ensured for Helium [5] and Ammbr [6], we still require it in our solution. Indeed, in this way, the users and local actors of the platform will then be authenticated by their respective home operators in a way that would preserve their privacy.

Every player participating in BALAdIN will then be trustworthy, as the Blockchain will then be permissioned (hybrid). Furthermore, local actors could provide some Quality of Service (QoS) and Quality of Experience (QoE), thanks to the help of their telco.

From the Blockchain side, the authentication system is very simple: each operator generates a ledger identifier for their customers/local actors and devices. Within the Blockchain, these identifiers are certified with a simple Public Key Infrastructure (PKI) mechanism where each operator would act as a trusted root authority.

In our solution, the PoB is not used as a consensus mechanism. PoB frames are simply stored into the Blockchain using PoB transactions, as a way to keep track of each network path and their use in a decentralised way. Using PoB will allow us to set up a bandwidth allocation and billing system similar to PayFlow [7].

Also, there are no assignment servers as the paths are self-regulated. Indeed, unlike Torcoin, the path nodes are physical devices (smartphone, base stations, etc.). Therefore they cannot be shuffled. However:

- Operators want their resources to be used and optimised; therefore on such a multi-actor collaborative platform they should provide a way to measure traffic in a decentralised way. In the case of the visited operator (Green); this system will ensure a fair measure of traffic used by others and the home operator (Yellow) will then be able to use this decentralised mechanism to testify its usage of the collaborative path. This protocol will then strengthen inter-operator roaming agreements thanks to the induced trust.
- The Local Actor (Alice) needs incentives to deploy cells; therefore a fair remuneration based on the use of them will be set up for them. A foolproof traffic measurement mechanism will also give them the necessary trust.
- And finally, the customer (Bob) is also part of the PoB process. While it is still possible for him to block the frame and cheat on the measure, the path could in return be killed by the local actor or the operator.

As a result, we believe that the collusion risk is low in our case, as it would not benefit any of the stakeholders.

While these assumptions are not sufficient in the case of a fully public Blockchain, in our case the use of a Permissioned Blockchain may prevent any risks. Telcos act as trusted third parties, to grant the necessary trust on the users/local actors.

While fraud detection mechanisms themselves are outside the scope of our paper, we suggest that Telcos may use such mechanisms to blacklist cheating and colluding users.

Let us now see in detail how our solution works.

3.1 Path creation

The “path creation” mechanism is described in Figure 2. Initially, Bob and Alice’s identifiers are both registered on the ledger by their respective operator Public Key Infrastructure (PKI).

Bob’s User Equipment (UE) will at first attach itself to Alice’s small cell using standard protocol such as mobile roaming or Wi-Fi, and then send an attachment

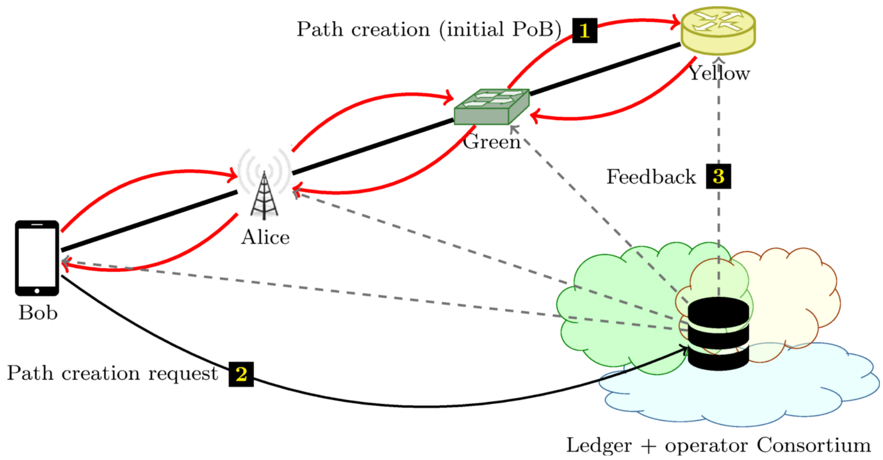

Fig. 2 BALAdIN path creation.

request by generating an initial PoB frame. The frame is first fed by this request and will be forwarded to the Green operator's Gateway, then to the Yellow Router (step **1**). Therefore, given that:

– Alice and Bob are both registered by their operators on the Blockchain;
– roaming is possible from Green to Yellow operator; and
– each operator agrees on the creation of such a path

; the PoB frame will then come back all the way down from the Yellow router to Bob. Bob's UE will then generate an "attachment" transaction that will be submitted on the Blockchain (step **2**). This transaction shall then trigger a smart-contract responsible for the activation of the path. This is step **3**: the transaction has been processed, and as a result, the nodes receive "positive feedback" from the ledger that will trigger the opening of traffic flow.

The "path creation" transaction will then be validated and its identifier stored on the Blockchain. The transaction will then be accepted by the validator nodes only if Bob and Alice's agents have been registered, the identifiers used by Yellow router and Green switch agents are valid, and the initial PoB cryptographic variables are correct. Once the path is considered active, Bob may begin to use it.

3.2 Path flow

Once the path is created and active, traffic through it will be counted by every path user.

At a specific rate, a PoB frame will be instanced and sent by the client (Bob), containing information about the traffic exchanged. This frame will then perform a round trip on the path. During this round trip, the other path nodes will then compare their measurements to Bob's, and validate it using the process described in Section 3.4.

In our implementation, we believe that the PoB rate should be time-based, and not data-based, to ensure a regular flow of transactions for the ledger. Furthermore, such a mechanism would also make it possible to measure low-bandwidth nodes.

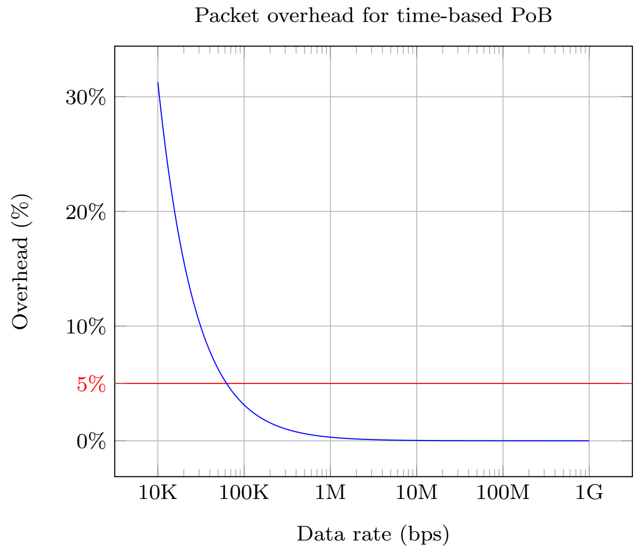

Fig. 3 BALAdIN PoB packet overhead

Let us now calculate the required PoB rate.

Given that:

- Most recent active mobile network peak rates exceed 1 GigaByte per second [19], we then assume an effective maximum rate of 1 Gbps;
- Thanks to the low cost of broadband nowadays, modern mobile broadband providers like Google Fi [20] bill each GigaByte of data.

Calculation to the nearest consumed GigaByte should be efficient enough for traffic measurement, for in case of a failed PoB transaction the financial loss would be minimised. Therefore, in our study the time between two PoB frames should be of $8s$, as the resulting precision would be of at least 1 GigaByte.

The packet overhead is then given by the following equation, plotted in Figure 3:

$$O_h = \frac{L_{PoB}}{\Delta_t B_{ps}}, \tag{1}$$

where O_h represents the overhead, B_{ps} the path bit rate, Δ_t the time between two PoB frames (8s) and L_{PoB} the size of a PoB ($3.128kB$ in our case described in Section 3.4, using regular ciphering suites).

Although the packet overhead increases as the data rate decreases, this choice should not be an issue, as for a maximum data rate of 100 Kbps the overhead is less than 5% (Figure 3). And this rate has been overcome since EDGE networks were deployed.

How the PoB agents should count the traffic is still a question that needs addressing, for multiple factors are at stake (up/down traffic, the wide variety of protocols being used nowadays in mobile networks, etc.). Moreover, depending on the measurement mechanism, the different measures may diverge on a BALAdiN path.

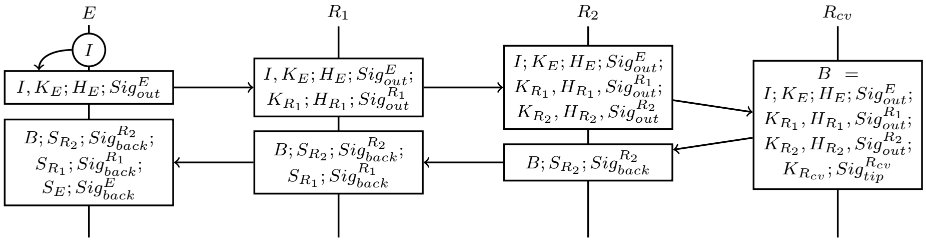

Fig. 4 The PoB process

As a result, PoB agents should use a set of simple rules such as thresholds for measurement comparison, in order to reach an agreement regarding PoB, and successfully perform the whole PoB mechanism.

Then, similarly to the "path creation" (Figure 2), a PoB transaction will be sent to the ledger, and path users will monitor its successful commitment.

3.3 Path destruction

Any path user that detects an attempt to cheat should shut the path down and send the incomplete frame to its operator for investigation.

Relays should also implement a "timeout" mechanism, to terminate the path if PoB are not validated and committed onto the ledger within the limits, or if a PoB transaction is found by the validator nodes to be invalid.

The network path is at first directly terminated, then the event should be logged onto the Blockchain using a "path destruction" transaction. This transaction may be sent any time, by any path actor (User, Local Actor, or any operator).

3.4 A closer look at our Proof of Bandwidth Implementation

The process is described in Figure 4. To simplify the issue, Bob's UE is considered the **Emitter** (E), as it starts the process, the Yellow Router the **Receiver** (R_{cv}) as it is the target that Bob wants to reach; and Alice and Green switches are considered **Relays** (R_1 and R_2), as they relay traffic and **provide network resources**.

At first, the Emitter creates the PoB initial frame I containing:

- The PoB number $P\#$; For the initial PoB, $P\#$ is set to zero;
- The timestamp in POSIX64 format t;
- The Path identifier P_{id}. This identifier is calculated thanks to a hash function performed on the path users' identifiers.
- The padding (information for path users about the next PoB time), Π (default to 8s);
- The amount of data exchanged since the last PoB, Λ.

Hence, let $I = \{P\#; t; P_{id}; \Pi; \Lambda\}$. This frame will then hop through the network path, performing a round trip. At each step (relay 1, ...), the other path users (relays, receiver) will, upon approval of I, perform cryptographic calculations as defined in [11], and digitally sign the frame. These signatures and variables will then carry a “Proof of Approval” of the PoB frame, and shall then guarantee the integrity of the data contained into I.

On the contrary, each actor within the path should ignore any PoB frame that they find corrupted, and consider ending the path, in accordance with the process described in Section 3.3.

Compared to Torcoin, some additions have been made on the PoB:

- The frame is actually signed in both directions by users;
- The initial tuple (I) is improved, by the addition of extra fields:
  - The amount of data exchanged on the path since the last PoB;
  - The timestamp corresponding to the creation of the initial tuple;
  - A field indicating when the next PoB should occur (either in data or time unit);
  - And a path identifier (set to zero for the first one, as it is unknown).

These changes are motivated by the context of the use of this mechanism. Indeed:

- The double signature will allow everyone on a path to report abuse to its operator, and the incomplete PoB will help to identify where the failure has occurred;
- Logging the data amount/the timestamp of the initiation of the PoB will allow more precise metrics, as a given path throughput could then be measured.

Although we added more complexity to the frame, we believe it would neither be an issue for resource consumption nor byte overhead, thanks to the relative simplicity of current cryptography algorithms such as RSA for digital signatures/SHA256 for hashes.

In order to confirm this assumption in our case, we performed a benchmark of the SHA 256 and RSA 2048 performances on “basic” Virtual Machines (2GB RAM/2 virtual CPUs), using the `openssl` tool.

```
### SHA256 ###
Doing sha256 for 3s on 16 size blocks: 5130762 sha256's in 2.99s
Doing sha256 for 3s on 64 size blocks: 2770388 sha256's in 3.00s
Doing sha256 for 3s on 256 size blocks: 1220497 sha256's in 2.99s
Doing sha256 for 3s on 1024 size blocks: 362161 sha256's in 3.00s
Doing sha256 for 3s on 8192 size blocks: 47681 sha256's in 2.99s

### RSA 2048 ###
                     sign     verify     sign/s verify/s
rsa 2048 bits 0.002936s 0.000088s     340.6   11400.8
```

Fig. 5 RSA 2048 and SHA256 performance.

The results, in Figure 5 show that the cryptographic operations performed to create a PoB should be doable on every kind of device, as low-end smartphones now have the tested configuration (2GB ram, dual-core CPU) as standard.

3.5 Transaction validation on the Blockchain side

For any received PoB transaction, the validator nodes should carefully check that:

- The path users are both registered on the Blockchain and are active;
- The path is active (has been properly enabled with initial PoB, and has not been destroyed by any user), except in the case of the initial PoB;
- All cryptographic variables (hashes, signatures) are valid;
- The data contained in I is realistic (the $P\#$ is incremented properly).

As the traffic exchanged on the path is opaque from the Blockchain side, the integrity of this metric should then be validated by the path users themselves, which implies methods to ensure reconciliation. However, we believe the result of the process is trustworthy enough, as a PoB frame requires everyone's approval in a consensual way.

4 BALAdIN performance evaluation

4.1 Limitations of Blockchain for our use-case

When dealing with Blockchains one must bear in mind the limitations of such a model, as peer-to-peer databases do not behave like centralised ones. These limitations are mainly related to limited scalability and transaction propagation/commitment. The first limitation is that Blockchains are known to have a limited global transaction throughput, known as Transactions Per Second (TPS), and to not scale very well.

Let us then calculate the required TPS for a BALAdIN network. Let $T_{PS} = x/\Delta_t$, with x the number of simultaneously active paths, and Δ_t the time between two PoB frames, as each PoB frame is associated with a PoB transaction.

The maximum amount of simultaneous active paths can be then defined by $x_{max} = T_{PS_{max}} \times \Delta_t$ ($\Delta_t = 8s$).

The maximum transaction rate of a Blockchain varies from implementation to implementation. At the time of our study, the most scalable Blockchain is Hyperledger's Sawtooth project [21], for it implements a novel consensus mechanism called the "Proof of Elapsed Time" [22]. This implementation allows a theoretical maximum TPS of 1000.

On a Sawtooth Implementation of the BALAdIN PoB, there would then simultaneously be a maximum of $1000 \times 8 = 8000$ active paths.

However, the number of expected simultaneous users is difficult to assess, for it would depend on the level of coverage (regional? national? worldwide?) of a BALAdIN network. For our study, we used the statistics of public Wi-Fi hotspot coverage nowadays, as it is close to our use-case.

Some projects like World WiFi [23] have estimated that there are more than 4.5 million hotspots worldwide. Therefore, given that World WiFi estimates a simultaneous average of 2-3 users per hotspot, the excepted global transaction rate would be of $1,125,000$ - $1,687,500$ transactions per second. Therefore a worldwide unique blockchain is not doable in our case, for it would require a very fast ledger.

Regional coverage seems more doable. As the O2 service has $15,000$ hotspots through UK [24], the expected number of simultaneously active connections through the entire UK can be estimated to $45,000$. The UK could then be split into 6 separate regions with their own BALAdIN networks and Blockchains.

As the PoB transactions of two separate, distinct paths should not interfere, deploying multiple local Blockchains should be doable in our case.

Along with speed comes also the question of the commitment and validation of transactions on a Distributed Ledger. Indeed, whereas reading data on a Blockchain is not a problem, committing transactions is much harder. The following questions must be addressed in order to make such an application work, and to make users trust the ledger:

– When a given transaction has been approved by a validator node, what is the probability of its approval and commitment by the others nodes?
– How much time does a transaction need to propagate through a validating node network?
– And finally, how can a given transaction be considered as committed?

These questions may be hard to answer, due to the peer-to-peer nature of a Blockchain Application. Transactions are indeed processed by every individual validator node, and their global commitments are never final, as they can be ignored and rejected by the other nodes of the network at any time. Furthermore, a Blockchain may experience a *fork* at any time: at some point two distinct blocks of the same height may be generated at the same time, thus creating two separate blockchains from a single block. When such an event occurs, the validating node should select the longest branch and discard the blocks of the other branch, thus invalidating their transactions. The other branch, then composed of "stale blocks", is ignored and its transactions discarded. The consensus mechanisms in place within Blockchains are designed to handle such failures, so that it is hard for an adversary to cancel (and then double spend) a registered transaction by forking the chain. [25] suggests that the confirmation of a given transaction should be limited to a probabilistic statement, as the more blocks confirm a given transaction, the less likely it is to be rejected [26]. There must then be a compromise between transaction validation time and the probability of its rejection: for security-sensitive applications such as transactions implying valuable assets, it is advised to wait for more confirmations than for non-sensitive, low-value transactions.

4.2 Modelling PoB integration onto BALAdIN

We now model the peer-to-peer network hosting the Blockchain supporting and handling the PoB, to asses the Blockchain-side performance regarding the maximum transaction throughput, and the propagation of the transactions within a small network to scale our system appropriately.

Our results will also allow us to come up with a deployment scheme for validator nodes, with a proper PoB rate within a path, a realistic "timeout" for PoB validation, and to know how validator nodes should consider a PoB transaction, as these steps depend on the observed PoB transaction validation and propagation time.

For our modelling purposes, the Hyperledger Sawtooth is the most basic Blockchain engine we have found. Indeed, the Sawtooth Application Programming Interface (API) allows any application to interact with the Blockchain like a regular database: programs called "transaction processors" are deployed on validator nodes. These pieces of software are responsible for handling and decoding transactions, validating them or not, and updating an abstract representation of the ledger content called the "State Database". This is the behaviour we need, as Sawtooth manages all Blockchain-specific operations (peering, consensus mechanism, block structure, etc.) while the actual data stored onto the ledger is application-specific. Moreover, the Hyperledger Sawtooth Blockchain proposes a Proof of Elapsed Time (PoET) for block validation, a consensus mechanism known to be fast [22].

As depicted in Figure 6, we have thus created a transaction processor for handling our PoB transactions. For development purposes, the transaction processor is designed to handle not only "path creation", "path destruction' and regular PoB transactions, but also user ("client" and "relay") registrations.

All the data necessary to handle BALAdIN transactions (registered users' keys, active paths and their parameters) are thus stored in the State Database. Using it then allows the transaction processor to fully check for the integrity of the PoBs, as described in Section 3.5.

We have also created programs to send mock PoB transactions to the validators. These programs simulate every user in a mock BALAdIN path in order to test the transaction processor and the Blockchain as a whole, and follow this process:

- As the first stage for each simulated simultaneous path, they generate the necessary keypairs (the two clients, and the two relays).
- Then, using previously generated keypairs, they submit the initial PoB transactions described in Section 3.1 to register the paths.
- Then, at a specific rate, they submit the PoB transactions for each simulated path.

We have created two networks of nodes, separated by a router, as depicted in Figure 7. The goal of this router is to emulate realistic network propagation delays between two distinct areas by adding latency with the help of the `tc` command [1].

For each subnet:

- The three nodes are monitored:
- On each side, two nodes are "active", as they receive transactions to commit from the test programs, and the third node is only "passive" (does not receive any submitted transactions).

We have chosen to deploy an Openstack suite on a single-node (Devstack), as it allowed us to get the simplest installation, with the use of basic and low-level tools such as openvswitch for networking, qemu/KVM for virtualisation, etc.

To assess the performance of the Blockchain we generated mock PoB transactions and submitted them to validator nodes at a specific rate, and with the central router simulating a "realistic" network latency, following a random normal distribution centred at 50ms.

We chose to deploy our validators on virtual machines with a dual-core CPU (2 VCPUs) and with 2GB of RAM, which corresponds to the hardware of a nano-computer nowadays.

We also assessed the propagation time of our transactions by the use of "probes":

- At first, the nodes are synchronised on the same clock using an external Network Time Protocol (NTP) server;
- One active node of our testbed generates and submits PoB transactions with a special "flag", and logs the submission time;

[1] https://linux.die.net/man/8/tc

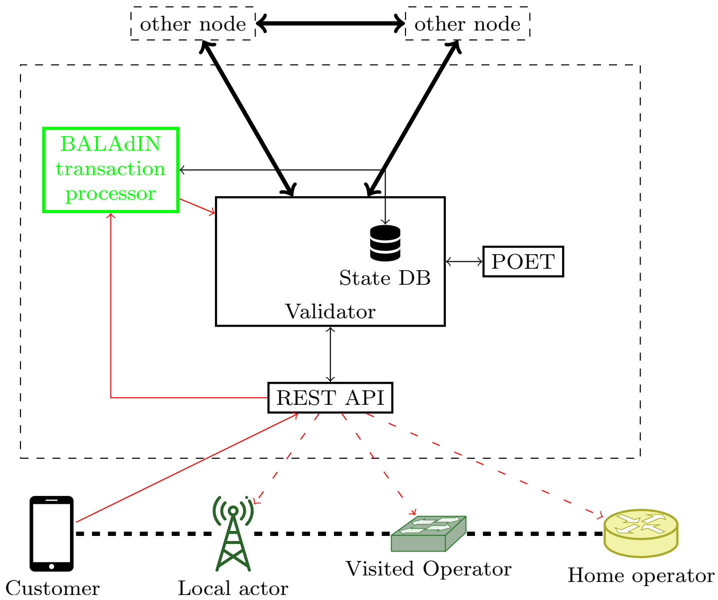

Fig. 6 BALAdIN sawtooth validator node. While the core of the validator, the consensus mechanism and the REST API, are provided by Sawtooth, the transaction processor and Tx submitter Agent deployed on the customer UE are of our own design. The Validator schedules the new blocks; manages the State Database and manages the interconnection with other nodes. With Sawtooth, the consensus mechanism is actually implemented in a separate component. The REST API acts as a proxy for incoming transactions by providing convenient ways of submitting transactions and monitoring the chain via a REST API and a WebSocket server. In our case, the local actor, visited operator and home operator will subscribe to the WebSocket channel to monitor the commitment of the PoBs.

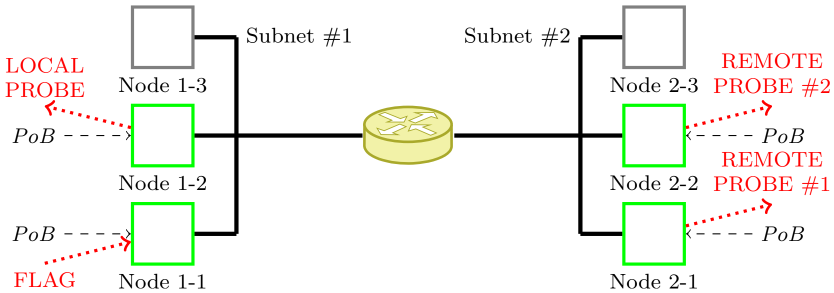

Fig. 7 BALAdIN network testbed. “active” nodes are in green, and “passive” ones are in grey. The network is split into two subnets separated by a router to simulate the two edges of a given network, by eventually decreasing the router network performances.

– “probes” are set up on the three other active nodes. These pieces of software monitor the incoming Blockchain transactions, identify the “flagged” ones, and log the reception time.
– The measured transaction propagation time here is, for each probe, the difference between the submission time, and the reception time of a “flagged” PoB transaction.

We conducted this experiment when the Blockchain was idle (no other PoB submitted), and then at a transaction rate of 10 as shown in Figure 8.

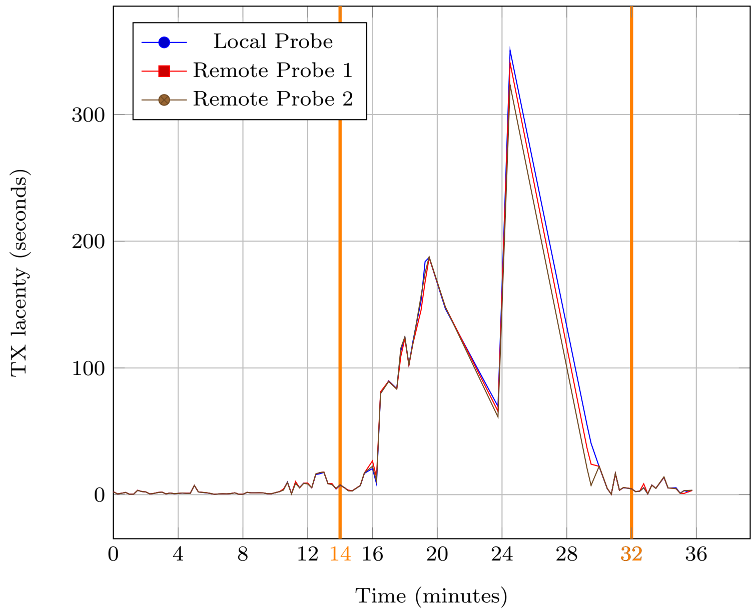

Fig. 8 PoB transaction latency on a "realistic" validator network (random delay induced between the two subnets, following a normal distribution centred at 50ms). The test was performed with the Blockchain idle (0 to 12 minutes), then with a load of 10 TPS (12 to 29 minutes), then idle again (after 29 minutes). When the Blockchain is idle, the transaction propagation time was relatively low (2-3 seconds). However, under load, this propagation time dramatically increased to up to 5 minutes.

The results shown in Figure 8 show that the transaction propagation time increases with a big load of transactions per second.

Through these experiments, we showed that the validator network global throughput has a strong impact on the transaction propagation time.

5 Deployment of a fast Blockchain for the Edge

As depicted in Section 4.1, a globalised, general deployment (full nodes everywhere) of our Blockchain is not feasible in our case, due to relatively high transaction rate.

Yet deploying multiple "local" Blockchains to handle PoB transactions should be achievable in our case, for two separate BALAdIN paths won't be conflicting in their process. However, some mechanisms need to be set up to prevent any user from simultaneously using paths from two separate regions, for this may lead to double spends afterwards.

5.1 Possible deployment

For better reactivity, deploying validator nodes right at the edge of the network, directly on the local actors's small cells seems to be a good solution. However, as shown in Section 3, such devices might not have the hardware required to process transactions, for under load the transaction propagation delay increases dramatically.

Some pre-processing could still be achieved at the edge, as cells managing multiple paths simultaneously could pack multiple PoB onto a single batch for faster processing for validator nodes. The integrity of the PoBs may also be checked at this stage, as the from application side this step is not too resource-hungry.

The validator nodes could then be deployed by operators at a regional level.

Such a hybrid approach is not fully decentralised. However, it will allow a better user experience for both customers and local actors, while preserving some levels of transparency for them. Furthermore, a reduced number of nodes will reduce the probability of desynchronisation and divergence of the Blockchain. Indeed, users and local actors of the platform would still be able to deploy validator nodes themselves, by providing the necessary hardware, so as to provide some transparency on the processing of PoB.

In that specific case, the latency induced by the transaction submission itself to a node in the region should not be a problem as it will be pretty low compared to the transaction commitment time.

Bearing this deployment in mind, a transaction may be considered as committed with zero confirmation thanks to the separation between paths, and the limited number of nodes. However, after too many failed submissions of PoB transactions, path users should consider destroying the path following the process described 3.3. How many are “Too many failed submissions” is still an open topic, for this should be determined by the underlying contract and the valuable assets involved. All submitted PoBs should, however, be cached by the emitter after submission. Indeed, as within a path, PoB are sequential (each PoB transaction is dependant on the previous one), they will need to be resubmitted if the frame is dropped anywhere in the network during the process. We therefore propose the deployment depicted in figure 9:

- An “overlay” system shared only between Telcos that would be responsible for managing user authentication, and the remuneration model. The structure this system would take remains an open topic as we have focused our work on the PoB integration. Yet a consortium Distributed Ledger can achieve this step, as it has been done before with Ammbr and Helium.
- Multiple “local” hybrid Blockchains, that would store PoB to keep track of the access network and user interactions. These Blockchains would interact with the “overlay” system in two ways:
  - With a PKI-like mechanism to authenticate BALAdIN users so that only registered users could use the system;
  - The remuneration engine would read the committed PoBs as a proof of the usage of given network paths; so that fair retribution/billing could then be ensured for every local actor/user.

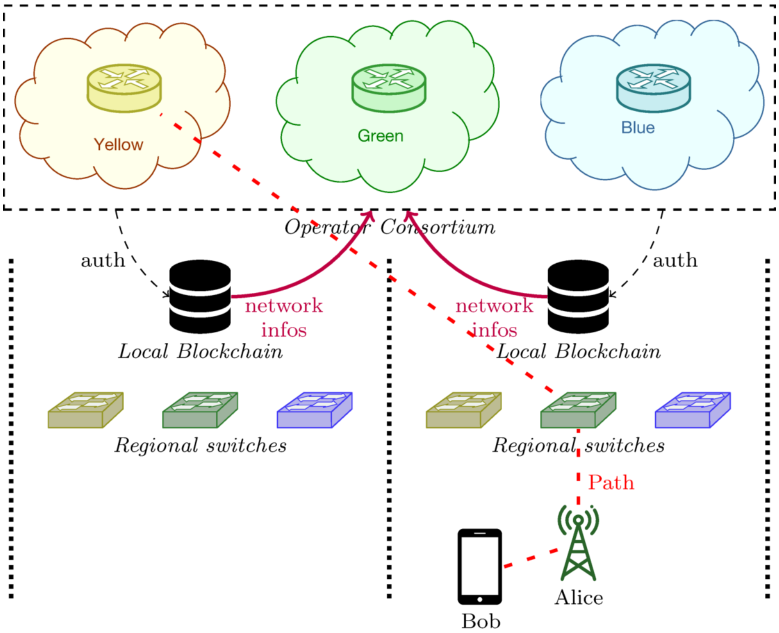

Fig. 9 Deployment of BALAdIN, involving a centralised trusted environment interacting with multiple local Blockchains used to collect PoBs

5.2 Open Issues / Future work

Although we have been able to asses the behaviour of the Blockchain holding the PoB, thus some of the key requirements of BALAdIN and its PoB integration, our model has some limitations that should be addressed in future experiments:

- At first, it is hard to efficiently simulate a Blockchain network using only virtual resources. Indeed, physical hosts still have limitations to successfully emulate a high number of virtual machines, such as disk/net IO
- Furthermore, for the Sawtooth case, the PoET mechanism should use the Intel' Software Guard Extension hardware feature [22]. As it then requires close interaction with the CPU hardware, it is impossible to run such a feature on a Virtual Machine. Whereas Sawtooth provides a "PoET simulator" consensus model to emulate this hardware requirement, the resulting computing requirements may decrease the overall performance.

Also, many components/functionalities of our solutions have not been covered yet:

- Authentication. This topic is a challenge, as such a system will need to preserve the privacy of each telco customer.
- The business model to implement on our platform to trade connectivity in the most efficient way, taking advice from similar systems like Ammbr, Helium, etc.
- The necessary safety mechanisms to provide the QoS/QoE required for end users. Indeed, as no telco holds the infrastructure as a whole, this topic requires particular attention.

– The integration of the PoB frames into the network needs to be discussed as well. Furthermore, due to the wide variety of protocols used in modern mobile networks, the question of which accounting/reconciliation mechanisms to use in the PoB process needs to be considered as well.

6 Conclusion

In this article, we presented BALAdIN, a Blockchain-based collaborative access network solution. The true potential of our solution lies in the use of a "Proof of Bandwidth" mechanism to monitor the usage of a network path by its users within a shared consortium network. We therefore avoided the introduction of an Oracle on our decentralised application, thanks to the trust provided.

We then implemented the Blockchain solution that holds these proofs, to make a model test network on virtual machines. We were able to uncover many challenges and issues relative to the implementation of a fast and scalable Blockchain. We proposed a solution with multiple local Blockchains to balance the processing of the PoBs.

During our studies, we uncovered the main limitations of the Blockchain for BALAdIN, by evaluating transaction throughput and propagation. Using our results we were able to come up with a possible deployment of our Blockchain. We have shown the need to deploy multiple Blockchains for processing PoBs, as such a mechanism is almost impossible to implement on a single Blockchain due to its limited throughput and scalability.

While the state-of-the-art Blockchain-based DLTs might not yet be mature enough to sustain future collaborative network architectures and new collaborative models such as BALAdIN, we have been able to uncover some of the features of our solution. In this publication, we have shown the need to limit the size of the infrastructure supporting the PoB mechanism. The best way to proceed is to run the PoB mechanisms over collaborative networks of limited scale. We are thus confident in the ability for DLT to support new collaborative network architectures in the future on the condition that their deployment be carefully monitored.

References

1. S. Nakamoto, et al. Bitcoin: A peer-to-peer electronic cash system (2008)
2. G. Salviotti, L.M. De Rossi, N. Abbatemarco, in *Proceedings of the 51st Hawaii International Conference on System Sciences* (2018)
3. G. Wood, et al., Ethereum: A secure decentralised generalised transaction ledger. Tech. rep. (2014). URL http://gavwood.com/Paper.pdf. Ethereum Project Yellow Paper
4. F. Armknecht, G.O. Karame, A. Mandal, F. Youssef, E. Zenner, in *International Conference on Trust and Trustworthy Computing* (Springer, 2015), pp. 163–180
5. A. Haleem, A. Allen, A. Thompson, M. Nijdam, R. Garg. Helium a decentralized wireless network (2018). URL http://whitepaper.helium.com/
6. Ammbr whitepaper (2018). URL https://www.ammbr.com/download/whitepaper/
7. D. Chen, Z. Zhang, A. Krishnan, B. Krishnamachari. Payflow : Micropayments for bandwidth reservations in software defined networks (2019)
8. Softbank, sprint, far eastone and tbcasoft launch blockchain consortium for telecom carriers. Sprint Newsroom (2017). URL https://newsroom.sprint.com/softbank-sprint-far-eastone-and-tbcasoft-launch-blockchain-consortium-for-telecom-carriers.htm

9. A. Adhiappan, A. Chernetsov, M. Fenomenov, U. Karabudak, A. Korabanova, S. Kislyakov, L. Le Beller, M. Nati, B. Radier, A. Sushkov, A. Ustimenko, A. Vedin, O. Yurlov, T. Ben Meriem, V. Messié, N. Omnes. Federated CSPs marketplace : A DLT-based data trust enabling business assurance for CSPs platforms federation (2020). URL https://www.tmforum.org/vertical-industry-telcos-federated-dlt-based-marketplace/
10. CSP use cases utilizing blockchain. TM Forum Technical Repor TR279, TMForum (2019). URL https://www.tmforum.org/resources/technical-report/tr279-csp-use-cases-utilizing-blockchain-v3-1/
11. M. Ghosh, M. Richardson, B. Ford, R. Jansen, A torpath to torcoin: Proof-of-bandwidth altcoins for compensating relays. Tech. rep., NAVAL RESEARCH LAB WASHINGTON DC (2014)
12. R. Dingledine, N. Mathewson, P. Syverson, Tor: The second-generation onion router. Tech. rep., Naval Research Lab Washington DC (2004)
13. V. Messié, G. Fromentoux, X. Marjou, N. Labidurie Omnes, in *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)* (IEEE, 2019), pp. 201–205
14. H. Zhang, N. Liu, X. Chu, K. Long, A.H. Aghvami, V.C. Leung. Network slicing based 5G and future mobile networks: mobility, resource management, and challenges. IEEE Communications Magazine **55**(8), 138 (2017)
15. H. Li, G. Shou, Y. Hu, Z. Guo, in *2016 4th IEEE international conference on mobile cloud computing, services, and engineering (MobileCloud)* (IEEE, 2016), pp. 83–84
16. Y. Mao, C. You, J. Zhang, K. Huang, K.B. Letaief. A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys & Tutorials **19**(4), 2322 (2017)
17. J. Cieplak, S. Leefatt. Smart contracts: A smart way to automate performance. Geo. L. Tech. Rev. **1**, 414 (2017)
18. D. Siegel. Understanding the dao attack. CoinDesk (2016)
19. J. Wannstom, LTE-advanced. Tech. rep., 3GPP (2013). URL https://www.3gpp.org/technologies/keywords-acronyms/97-lte-advanced
20. Google fi - a different kind of phone plan. URL https://fi.google.com/about/
21. Hyperledger sawtooth. URL https://www.hyperledger.org/use/sawtooth
22. L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, W. Shi, in *International Symposium on Stabilization, Safety, and Security of Distributed Systems* (Springer, 2017), pp. 282–297
23. World wi-fi – decentralized free wi-fi network powered by blockchain. URL https://en.worldwifi.io/
24. Free wi-fi anywhere | how to get o2 wifi | o2. URL https://www.o2.co.uk/connectivity/free-wifi
25. C. Decker, R. Wattenhofer, in *IEEE P2P 2013 Proceedings* (IEEE, 2013), pp. 1–10
26. I. Weber, V. Gramoli, A. Ponomarev, M. Staples, R. Holz, A.B. Tran, P. Rimba, in *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)* (2017), pp. 64–73. DOI 10.1109/SRDS.2017.15