SCTE·ISBE CABLE-TEC EXPO 2018 ATLANTA, GA OCTOBER 22-25

2018 Fall Technical Forum SCTE·ISBE • NCTA • CABLELABS

Solving All Our Problems… Sort of…

Blockchain Integrity, Security, and Reliability for Cable Use Cases

A Technical Paper prepared for SCTE•ISBE by

Steve Goeringer
Principal Security Architect
CableLabs
858 Coal Creek Circle, Louisville CO 80027
s.goeringer@cablelabs.com

Dr. Jason Rupe
Principal Architect, IEEE Blockchain Initiative Co-Chair
CableLabs
858 Coal Creek Circle, Louisville CO 80027
j.rupe@cablelabs.com, jrupe@ieee.org

Table of Contents

Title **Page Number**

Introduction

This paper surveys and categorizes blockchain use cases relevant to the cable ecosystem and then discusses key design factors in implementing appropriate blockchains. But first, it explains some basic principles and concepts of blockchains and distributed ledgers. The paper ends with some discussion about important concepts regarding these use cases, many of which are unique to blockchain.

Blockchain networks and distributed ledgers are explained through five important concepts:

- Authoritative history – blockchain networks have the ability to keep track of history with high integrity.
- Identity management and anonymity – blockchains are decoupled from identity management, but may require it.
- Event synchronization – distributed ledgers in a blockchain network can keep track of the order of events.
- Traffic flow measurement and message flow – blockchain networks can be used to enhance networking overall.
- Information reliability – once in the blockchain network, information is very difficult to change; but until it's in there, anything could happen.

The use cases explored in this paper can be organized into these categories:

- New and direct revenue – ways operators can generate new sources of revenue from this technology.
- Cost optimization – techniques to reduce costs of providing services using blockchain.
- Customer experience – methods to enhance the customer experience through use of blockchain.
- Reduce ecosystem friction – applications of blockchain that simplify what is otherwise complex.

To support these use cases, several important security and reliability concerns must be addressed:

- Identity, authentication, and authenticity of transactions and participants.
- Node and ledger distribution and redundancy.
- Network scale and performance considerations.
- Governance and code management.
- Attack vectors as a reliability problem.
- The meaning of reliability in the context of blockchains and distributed ledgers.

Readers are assumed to be familiar with blockchain topics. Background information may be found in [3]. This paper uses both the term blockchain and the phrase distributed ledgers. This use may seem synonymous; however, the authors view blockchain as a cryptographic network technology that produces distributed ledgers. This is an important distinction. Not all use cases require the relatively heavy assurances that blockchains entail but may still benefit from a distributed ledger.

What Good Are Distributed Ledgers and Blockchains, Anyway?

Before discussing use cases, it is useful to revisit the benefits of blockchains. This allows better determination of both how to design a blockchain for a given use case and whether a distributed ledger is necessary for that use case. A comprehensive approach for determining blockchain suitability for a given application is outlined by Scriber in [2].

Fundamentally, blockchains help to achieve security by design by providing highly secure logs of transactions. These logs are highly secure because they are distributed amongst many participants (making them hard to change without participants noticing), because the transactions they include have integrity assurances (signatures built in), and sometimes because they include additional cryptographic protections. However, they are not a replacement for other security controls. Distributed ledgers typically allow us to attest truth, not assert truth.

1. Authoritative history

Blockchains provide layers of integrity, as discussed by Goeringer in [1]. Typically, transactions are signed by clients when submitted to the blockchain. Transactions may include data elements (or anything that can be converted to a data element) or executable code (smart contracts) that may have additional integrity protections. Transactions (and their associated smart contracts) are validated by one or more types of nodes when they receive the transactions. These transactions are compiled into blocks, often using a process called a Merkle tree which really produces a hash of all the transactions included in the block. Blocks themselves are digitally signed by their processors (miners in Bitcoin and Ethereum, validators in HyperLedger). A given block includes the hash of the proceeding block in the blockchain (that's the feature that makes it a chain). The result of all these layers of integrity is a data structure that creates facts. We know that a transaction was submitted and that the contents of the transaction are (nearly) irrefutable. If transactions are signed (for example, using asymmetric keys), we can prove a given client address submitted the transaction.

Integrity seems intuitively useful, but it can be transformational. Many businesses, including multi-service operators, are complex. Network operations and service delivery can include many stakeholders, some independently responsible for their own profits and losses. Moreover, new business models may leverage multiple businesses in service delivery (in a network context, this may include multi-tenant virtualization). Inevitably, each stakeholder will maintain their own service records. Different measurements or even errors in those records may make data reconciliation difficult or even impossible.

A distributed ledger with high transaction integrity can provide an authoritative history of transactions. This can streamline business operations, with particular impact on processes impacting compliance (such as privacy management). But how do we enhance transaction integrity?

2. Identity management and anonymity

Some security pundits have suggested that blockchains remove the need for identity management. Many articles have stated that blockchains create trust. Neither of these statements is true.

Blockchains can be designed to support anonymous transaction submissions, or transactions decoupled from attribution. They can also be designed such that each transaction is submitted only by credentialed clients. For that matter, transactions need not be sourced by or attributed to a person or organization. A

given blockchain platform, in fact, may support either option. The fundamental question is whether a given use case requires the identity of submitters or not. If it does, a process must be applied to issue identity to submitters. This identity may, for instance, be applied to a wallet in the case of a digital currency or security. In this way, blockchains don't eliminate the need for identity management; rather, they consume or rely upon identity management. In contrast, if a given use case doesn't require the identity of submitters to be known (as, for example, in the case of Bitcoin), then identity isn't needed because the use case doesn't require it. The use of blockchains does not protect or remove identity.

In considering what this means for trust, it is important to reconsider the previous section again. Blockchains allow us to create histories of transactions in which submitted transactions can be treated as a fact. Rather than creating trust between entities, blockchains create a data structure in which entities can trust. If applicable to a given use case, this actually removes the need for trust.

However, we must be careful. The fact of the transaction does not mean the submitted transaction itself is accurate. It may have been tampered before it was signed. Or the terms submitted by one party are not actually the terms agreed (offline); the transaction submission did occur, but it is not accurate to what the parties negotiated. The point being that accuracy, authentication, and authorization features must be designed-in using identity management practices related, typically, to public key infrastructure solutions such as cryptographic signing and co-signing. Such systems cannot really be anonymous, of course.

3. Event synchronization

A byproduct of how blockchains work is that the integrity base they enable includes a high confidence ordering of events to a certain degree of granularity. The granularity relates to the length of time it takes to process a block. The result is that transactions included in different blocks can be time ordered with extreme confidence (e.g., ordinality can be treated as fact)[1]. For events (transactions) that are encoded within a single block, at least some uncertainty is introduced. Time (hacks) can be added to signed transactions to provide some level of confidence in event timing, but this returns us to traditional challenges in event timing (uncertainty in time distribution protocol, inaccurate or even malicious time codes by clients, etc.).

Moreover, as most blockchain implementations are distributed, high confidence event ordering can be visible to all stakeholders. The result is a highly useful mechanism for synchronization events in multi-stakeholder or similarly complex execution environments. This can be beneficial to managing workflows, tracking fulfillment, and possibly ensuring audit-ability for compliance or audit purposes.

4. Traffic flow management and message flow

Many use cases for distributed ledgers require use of a single kind of transaction for a simple, or at least consistent, purpose. This is the case, for example, with Bitcoin: all Bitcoin does, really, is track the distribution of spendable transactions between parties. However, more elaborate workflows can be developed that provide much more interesting capabilities. This can be done by programming specific behaviors at various kinds of clients. For example, different clients can be coded to process transactions in different ways. Another option, not mutually exclusive, is to use smart contracts. Smart contracts can be implemented in several ways, but the common approach allows conditional execution of transactions based on information provided in the transactions, including the identity of the parties the transaction goes between.

[1] A fork in a chain can record a conflict in this ordinality but should be considered a temporary anomaly as one possible reality will eventually be accepted as fact.

This allows blockchain networks to be means of transport for complex and conditional information between stakeholders. Moreover, complex rules can be applied against transaction transport. Unlike other transmission protocols (such as IP), strong integrity is designed into every transaction, individually and collectively, so very strong traffic and message flow reliability and security can be ensured.

5. Information reliability

Reliable information in the blockchain network will remain reliable, but unreliable information can equally be locked into a blockchain network's version of reality.

If reliable information enters the blockchain and propagates sufficiently, it will remain highly reliable as long as the blockchain is reliable. Once in the blockchain, the information is propagated through the network, and the result is that there is a large number of duplicate records of the transactions. When the nodes of the blockchain network are largely independent in risk and attack, the information in the system is going to be immutable with a high degree of certainty.

Nothing in blockchain architecture assures that information entered into the network is reliable, but there are ways to add some amount of assurance. Contracts can be checked for consistency with other code on the blockchain, and for reliable executability. Sources of information can be authorized and authenticated. Ownership and authority can be checked with information stored on other secured networks if not within the given blockchain network. But without these additional measures, unreliable information can be stored on a blockchain network easily.

There are known attack methods for blockchain networks which can introduce contradictory information, the equivalent of a double-spend in a smart contract implementation. But these attack vectors are very difficult to exploit as long as the system is kept in balance.

Many consider blockchain synonymous with permissionless systems, but that is not the case. From the perspective of information reliability, for a given use case, the full spectrum of permissioned to permissionless systems should be a design consideration [5].

Use Case Summary

Designing blockchains must be approached from the context of the use cases that may benefit from decentralized ledgers. There are literally thousands of blockchain use cases under various phases of development, but only a few relate well to the cable industry. These can be discussed concisely when organized into four categories: new and direct revenue generation, cost optimization, customer experience, and reduction of ecosystem friction. These are briefly discussed in the following subsections. Of course, there is overlap, and a given use case may apply to more than one categorization. Further, there will certainly be use cases that defy these categories. Readers are invited to use their own creativity, and to use the ideas here liberally.

1. New and direct revenue

Significant effort has been focused on whether distributed ledgers provide the basis for operators to enable new services or new markets. Digital currencies or securities may also allow generation of capital through initial coin offerings. Moreover, digital currencies may provide lower cost approaches to bi-directional transaction flows, supporting loyalty and reward programs. Some use case examples:

- Content focused coin offering – One of the most tangible digital assets the cable industry works with is, of course, content in the form of movies, television, and music. Tethering digital assets to

some form of digital currency may provide the basis for new payment models, including capital generation if a new security is generated using an initial coin offering (ICO).

- Games and eSports coin offering – Traditional content is not the only option available to operators. Many have trialed eSport and game related service offerings. Tying an eSport or game to a digital currency provides interesting options, including the opportunity to provide digital assets and new ways of adding non-traditional services into triple- and quad-play bundles.
- Digital goods provenance – Sales, fulfillment, and delivery of digital goods remain very attractive. Some operators offer opportunity to buy rights to digital assets. Use of a distributed ledger provides opportunity for digital goods provenance which may streamline ecosystem operations and provide an improved basis for trust to new entrants into this space. Operators enabling such capabilities may open new revenue opportunities while securing their roles in digital distribution in the future.
- Secure digital media – Related to provenance, providing secure digital media solutions in itself may provide value sufficient for revenue generation. This may be particularly true for user generated content where current methods of ownership assertion are insufficient. Blockchain technologies may also provide an opportunity to disrupt value chains on digital rights management.
- New model for ad revenue – The current advertising technology market is plagued with a variety of fraud and other security problems. It is also very complicated. Distributed ledgers that integrate ad delivery solutions and payment methods may streamline advertising technology while also providing better value to publishers, content owners, marketing firms, and advertisers through improved (but controlled) transparency.
- Multi-party billing – For some markets and market segments, the cost of participating in cable services can be perceived as very high. Providing a blockchain-based billing solution that allows multi-party billing to complex households, various forms of multi-tenant housing, and college campuses may allow much greater participation from those markets while ensuring the operator does get paid for service. This may apply particularly well to wireless access environments.
- Blockchain as a Service – It may not be practical for an operator to address all the potential new revenue that blockchain-based approaches may enable. Fortunately, blockchains can be designed as service platforms, able to support a wide range of transactions. Such blockchains can themselves be offered as a service.

2. Cost optimization

Multi-service operators are complex businesses providing a wide range of services over highly varied and also complex infrastructure. Any given access solution may have multiple stakeholders (CPE operations, access operations, access engineering, product management, security). Any given access solution may integrate perhaps dozens of vendors resulting in a wide range of interoperability and integration challenges. All this complexity inevitably leads to at least some inefficiency which means higher cost per unit served. Application of distributed ledgers may provide new ways to optimize service costs. Some cost optimization examples:

- Virtualization orchestration – Network function virtualization (NFV) is largely about achieving disruptive cost reduction by allowing use of general-purpose server infrastructure rather than “big iron” routing, switching, and CMTS solutions. It is also believed that NFV may support new information and computer technology business partnerships through multi-tenant and even multi-operator solutions. Orchestrating complex service chains may require authoritative history for billing purposes, strong identity management to prevent service theft, and event synchronization.
- Service authentication – Cable services have traditionally been largely premised focused: a given address is subscribed to a given bandwidth and set of features, and that’s that. As we move more

and more to over the top delivery, and households become more complex, more flexibility is necessary; but this has proven complicated. Improved ability to maintain histories between business units, coupled with tools to provide more complex traffic and message flow management, may provide better tools for service authentication.

- Dynamic service creation or provisioning (announce, publish, subscribe) – Traditional service creation and provisioning have been very manual, partially simply to double check all the records and tickets from multiple stake-holders. Blockchains may provide better tools for synchronizing service creation (possibly enabling full automation which has been an elusive goal for decades). Moreover, the distributed ledgers may provide completely new methods of coordinating provisioning activities, enabling much more flexible programmatic service delivery.
- Connectivity negotiation or transaction management – Smart contracts provide new ways to track customer opt-ins for service. Much more granular service agreements may be achieved through pervasive accounting and tracking of user agreements.
- Enhanced content protection – Long- and short-form content are experiencing serious piracy today, with significant impacts on the revenue of both content owners and cable operators. Moreover, ad fraud impacts the profitability of the entire ad tech industry. Even user generated content faces challenges and end users rely basically on the good will of the various services and sites that allow users to share their content. Blockchains provide new ways to assert ownership, track usage, and assert digital rights on content. Enhanced content protection may provide significant cost benefits to all content owners and integrity.
- Provenance – Supply chain integrity remains challenging, and particularly so in the realm of software. Development operations provides methods of achieving live builds and agile service delivery. However, it relies largely on both open and proprietary code dependencies that are hard to track. Blockchains may provide new ways to synchronize software builds, deconflict dependencies, and track both changes to codes and also who made those changes. All with unprecedented integrity.
- Scalable IoT – IoT is resulting in massive deployment of both independent, standalone devices and also intricate autonomous systems that blend IoT sensors and actuators with big data services. The result is explosive growth of managed and unmanaged deployment of devices to homes, businesses, enterprise, campuses, and communities. Manual processes cannot track and manage how all of these components will interact and interoperate. Operators need more dynamic security controls, more flexible on-boarding, adaptive service contracts, and new payment methods to deal with this growth cost effectively. IoT scalability will depend on all five of the benefit areas discussed above.
- Reputation-based authentication – Many large ecosystem operators (e.g., Apple, Google, Amazon, Samsung, etc.) and industry consortia (Open Connectivity Foundation) are working to ensure devices offered within their scope can securely access services. However, visibility between systems is minimal, and not all services and devices are part of these large ecosystems. Many solutions are developing to identify or finger print devices. Distributed ledgers may provide a common resource in home, business, and access networks to record device behaviors and apply trust decisions on authentication and network access. This provides the opportunity to assert reasonable network hygiene and keep costs of managing network security low.
- Media storage consolidation – Currently, most operator contracts for Video on Demand (VoD) services require an individual media copy for every concurrent use or view of the media. So, for example, if an operator wants to provide “Deathly Hallows” on demand to up to 10,000 users at a time, they may have to store up to 10,000 copies of that media on their servers. Distributed ledgers provide the opportunity to reduce the need for trust through better identity management and record keeping. If media owners are made more comfortable by removing the need for trust,

the ability to reduce the number of stored copies of media provide the opportunity for massive cost reduction in storage.

- VoD evolution – Similarly to media storage consolidation, usage rights available to operators can be very restrictive. Again, this is largely due to the need for cost prohibitive trust solutions that simply have not been possible before. Distributed ledgers provide for much more secure and visible transaction management that may provide for much more engaging use experiences while maintaining equity amongst the stake holders (operators, studios, content aggregators, subscribers).
- NFV Management – Orchestration of NFV-based service delivery includes many, many distributed components. Moreover, multiple stakeholders may be engaged (multiple tenants, multiple operators). Event synchronization and tracking using traditional mechanisms may be very difficult. Distributed ledgers may provide much more streamlined orchestration.

3. Customer experience

The benefits of blockchains described previously can provide the basis for streamlined assurance of customer experience. Moreover, the fundamental capabilities of distributed ledgers provide the opportunity to evolve customer experience. This is, of course, challenging, and so the list of examples is correspondingly less than shown for new revenue and cost optimization. Here are four:

- Customer preference tracking – Managing customer preference choices across multiple platforms can be challenging. Moreover, subscribers desire service mobility. And, all their choices are subject to privacy considerations. Distributed ledgers provide the opportunity for streamlined, seamless customer engagement. Also, distributed ledgers provide the opportunity to leverage crypto currency solutions, and so customer preference choices can be more easily coupled to billing.
- Customer loyalty activities – In many markets, the cost of cable-based services can seem very high. Lowering the cost of cable service (both actual and perceived costs) while enabling alternative revenues may be helpful to many subscribers. This can be realized through various customer loyalty activities tracked through distributed ledgers. This can include discounts for ad watching, credits for customer referrals or service recommendations, 3rd party partnerships, and multi-payer households.
- Customer as content provider – User generated content transforms entertainment from storytelling to story sharing – from a passive consumption of presented content to generation and sharing of our own stories. YouTube and Facebook are, of course, the epitome of current user experiences in customer generated content. However, both services make tough compromises in allowing users to control and own the content they share. Distributed ledgers provide the opportunity for users to register and assert ownership rights in ways that have not been possible previous. And, because of the strong ability to secure the records of transactions, operators that enable new content sharing options to users can enable entirely new experiences and control to users in how they share and distribute their content.
- Media sharing – Allowing users to share content amongst themselves has been problematic in many ways. Consequently, license rights on distribution simply have not allowed consumers to share media. The authoritative history provided by a distributed ledger, coupled with strong identity management, may provide the basis to enable media sharing. Event synchronization, coupled with complex transaction flows, can allow very intricate user experiences that improve value and increase engagement among communities of subscribers.

4. Reduce ecosystem friction

The ability to reduce complex transactions to a matter of fact provides a new basis for trust between stake holders. This provides the opportunity to reinvent entire industries. That does sound audacious. However, we have seen fundamental disruptions in transportation (Uber, Lyft) and hospitality (AirBnB). Why shouldn't cable experience similar transformation? Here are five examples:

- Distributed trust – Public Key Infrastructure solutions remain one of the most scalable tools to assert identity management across all the evolving ecosystems that comprise the world of information and computer technology, and the related emerging area of IoT. However, PKI is complicated and its use introduces challenging supply chain risks in the identity supply chain. The result is that many companies and organizations are all pursing development of independent PKI roots (the foundational private key that attests the identity of all the certificates in that ecosystem). Unfortunately, bridging PKI roots is complicated and can introduce additional security risk. Leveraging distributed ledgers to orchestrate certificate issuance may provide a highly secure (reliable, high integrity) means of allowing different ecosystems make trust decisions relying on certificates from other ecosystems.
- Content distribution convergence – Several of the ideas above addressed digital transformation of media distribution, usually in the form of video (movies, TV). However, what works for movies might work for books, audio, music, and maybe the evolution of these media to AR and VR. If so, this provides the opportunity for operators to enter other content markets, or to provide new value to those markets.
- Royalty management and reconciliation – One of the hardest entertainment industry challenges is ensuring all the contributors to great content get what they are owed. Royalty management and reconciliation have traditionally reduced to rule of thumb-based estimates that may, or may not, have any basis on reality. Distributed ledgers, with or without smart contract capability, may provide cost effective ways to create authoritative histories of distribution and viewing that allow complete transformation of royalty rights management.
- Customer as content provider – Cable service providers are in a great position to assure, from the edge through to the core, that a person who creates content can have assured ownership of that content. Using a blockchain network, a customer who creates content can assure the content is encoded into the block, and therefore securing ownership of that content. Equally, they can transact that content in various ways, perhaps transferring ownership. The service provider can provide the blockchain-based solution to secure intellectual property for the customers, thus providing evidence of invention, creation, and ownership. Of course, this is a double-edged sword: a customer who plagiarizes will equally lock the evidence in the same process that protects intellectual property.
- Media sharing – In a manner like described above, media can be shared and permissions managed on a blockchain network. The distinction here is that rather than manage all the details of a relationship as needed when the customer becomes the content provider, this use case is simplified so that media can be shared as the user intends, without ownership transfer, contracts, or asset exchange beyond the sharing of initial content.

Complex Security and Reliability Design Concerns

1. Identity, transaction authentication, and transaction authenticity

There is an important subtlety when considering the idea that transactions become facts when recorded on a distributed ledger using blockchains. The fact is that the transaction on the distributed ledger is the transaction that the client submitted. That's it. However, that does not mean that what the client submitted was what they intended to submit, nor whether the transaction submitted was what another client expected or agreed to. Moreover, without some access controls, the transaction submitted to the blockchain network may not actually be what the client sent.

Consequently, specific use cases may need additional security controls added. Two functional areas to consider are identity and authenticity.

Nearly all blockchains use some form of asymmetric key pair to prove ownership of a transaction. A private key is used to sign the transaction; a public key (often included in the transaction or even used as a transaction identifier) is used to decrypt the signature and prove that the transaction is authentic. However, this can be anonymous. If it is important that a given transaction be authenticated and authorized prior to inclusion on a blockchain, identity should (must, really) be issued by an identified authority. It may be possible to use some form of distributed organization to issue identity, but most commonly a PKI certificate authority is used. Then, whenever a client submits a transaction, the transaction will be signed by the client and will include its PKI certificate which is in turn signed by the PKI authorities. This provides a strong basis to attest identity of clients.

How can we assure that what one client submits is what another client has agreed? There are several means, but one is to co-sign the transaction. A signed transaction can be provided by one client to another who then can review and accept the transaction, sign it, and submit it to the blockchain network. Alternatively, both clients can submit transactions, and a validator of some sort can ensure they match prior to approving the transaction for a block. And, of course, some form of smart contract can be used.

Architecture can matter a great deal. For example, it may not be feasible to adequately secure keys for a client on subscriber owned devices. If those keys can be accessed or manipulated, transaction identities cannot be assured, and therefore authentication and authenticity are at risk. So, it may be desirable to use a proxy for the end client (for example, deploying the users' "wallets" on the cloud).

A final note on identity management is warranted. Many architects and solutions providers are attempting to use blockchains as an alternative to strong identity management. Identity must be attestable in some way. This usually requires some type of central authority (such as a certificate authority in a PKI). It may be possible to make some level of trust decisions based on behaviors of clients recorded by peers on a blockchain. However, past behavior is not always indicative of motivations, and therefore may not be indicative of future behaviors. Moreover, it is difficult for a reputation-based system to protect from Sybil attacks [4]. Consequently, it may be more prudent to consider blockchains as consumers of identity rather than proxies for identity.

2. Distribution and redundancy

Distributing the ledger of transactions is a design approach specifically to achieve fault tolerance. The nature of the threat here – that nodes and links can fail or actually be hostile – is a well-defined problem

known as the Byzantine General's Problem [6]. Today, many mechanisms have been crafted, mostly inspired by "Practical Byzantine Fault Tolerance and Proactive Recovery" [7]. Within blockchains, the common algorithmic approach to achieve Byzantine Fault Tolerance (BFT) is to use a consensus protocol. One of the earliest (perhaps the first) formal algorithms for consensus in computer science is Paxos [8].

While these excellent papers provide the formal definitions and approaches to achieving fault tolerance in an uncertain world, their notions can be simply described as "distribute authoritative copies of your transactions widely". The usefulness of the papers is to help understand how to determine how widely distributed and how authoritative is appropriate for a given level of confidence (e.g., security). Common wisdom is that we use a consensus approach designed to achieve at least 51% consensus, and that we need to have a certain minimum number of nodes to achieve tolerance to a certain number of faults (3f+1 in [6]).

However, this generalization may have some issues. Consider that a real battlefield has terrain – the ability of a given general to attack or defend or maneuver may be constrained. Further, consider that any given general may not be equal to others in terms of capability or forces. And finally consider that the situation of the terrain and the general likely change over time (for example, because of weather or time of day). In other words, practical BFT must be designed according to the specific conditions in which any given blockchain exists. Applying this idea to familiar concepts of blockchains, some miners (generals) may have higher hashing rates than others and be served by different scales of bandwidth, which in itself may be constrained (by a national firewall, for example). Furthermore, Internet performance varies over time because of global events and natural conditions.

In other words, BFT must be weighted according to the realities of a given blockchain. We may need many more nodes than 3f+1, or nodes may need to be constrained in some way to achieve a given security result. The closer a blockchain network is to the lower node counts, and the higher the number of faults (failed or malevolent miners/validators), the less likely the integrity of transactions and associated blocks during that period of time. Moreover, the more time it is, the more likely for the blockchain network to come to consensus. This, in turn, may result in uncertainty on the confidence of integrity or validity of a given transaction.

3. Network scale and performance considerations

More nodes in the network doesn't necessarily mean more reliability. For a defined level of consensus, a larger network will take more time than a smaller one, and some transactions are timely, so we need to continue under the assumption that consensus will be achieved, though not guaranteed yet.

As a network grows in scale, propagating a transaction across the network takes more time, and there is an increase in the probability that the blockchain will split. But a well-designed blockchain will handle these situations eventually. That means some amount of time is required to gain high assurance that the transaction becomes fact. That amount of time, for a given amount of certainty (risk), increases with network scale.

But performance may increase from a certain perspective, with the increase in network scale. As the nodes on the network spread farther and wider, access to the network increases, thus reducing the time required to put a transaction onto the blockchain. Performance of the initial step can therefore reduce. The next step is locking the transaction onto a block, as the message propagates. With more nodes, the speed of locking the transaction onto a block should reduce as more nodes compete. The propagation of the information to other nodes on the network should spread at roughly the same speed on a per node basis.

To see this result, consider a model that considers the time to propagate the message, and then to encode onto a block. The time to propagate to a given number of nodes is a function of the network connectivity, connection speed, and processing speed, which should not reduce with network scale (though in some cases it could). The time to encoding onto a block is an order statistics problem, in which the time to first encoding increases with participation. The net result is that in reasonable blockchain network designs, the time to lock a transaction onto the blockchain should reduce with an increase in network size, all other things being equal.

When validating a transaction, how much validation is enough, and how long can an application wait for the validation process? Consider a permissionless network, where participants can join and leave at will. There may be no control as to the membership of subgroups, or their ability to collude. Validation of a transaction among one aligned group of nodes is less assurance than validation by a large group of diverse, non-aligned nodes. In a permissioned network, however, all nodes may be aligned by design, and presumably trusted equally. A single validation may be nearly as good as validation across the entire network in that case.

It may seem that the larger the blockchain network, the greater the chance that a high degree of integrity per transaction can be achieved. This does not necessarily follow. The larger the network, the more likely it is that the network will include bad actors. Moreover, the larger the network, the more messages must be exchanged to achieve BFT (so bandwidth efficiency decreases), and the longer it may take for a given network to come to consensus.

Therefore, if transaction validation time and block confirmation are critical design factors, the size of the network and the associated bandwidth must be designed. In most cases, it will be seen that a smaller blockchain network will result in faster transaction processing times while the security (integrity) of transactions and blocks decreases.

4. Governance and code management

Blockchain networks are very complex systems of hardware, software, and people. Reliability best practices for each element should be followed, but that is not guaranteed. The severe redundancy of distributed blockchain networks is assumed to cover many failure modes, but the tradeoff is not always positive. Because of this complexity, the governance of a blockchain and the processes and practices of managing code become quite important. From a risk management perspective relative to any given use case or service, the body governing a blockchain and the method of how code is managed should actually be considered as supply chain risks and assessed accordingly.

We can consider a specific case in a fork of Ethereum. In June 2016, a major hack was operated against Ethereum that resulted in $70M of asset losses [9]. To “fix” the results of the hack, the Ethereum community decided (not unanimously) to return all the Ether stolen by changing the smart contract associated transactions. This basically was a willing hack of the immutability of Ethereum by its governance body, with a forced code update that was not accepted by all. Ethereum Classic was born out of the mess.

5. Attack vectors as a reliability problem

While it is beyond the scope of this paper to explore all the known or envisioned attack vectors for blockchain, it is important to note that blockchains must be designed to mitigate known attack vectors, and the list of these vectors is increasing every day. Considering security as a quality of blockchains, security becomes a potential mode of failure for a blockchain. As the attack vectors that apply to a given

blockchain network increase, the blockchain network experiences a form of aging, and as it ages, its rate of failure increases; over time, the blockchain becomes more vulnerable to security failures. Thus, stable blockchains, in relation to security failure modes, experience an increasing hazard rate. Redesign and upgrade are therefore necessary to sustain a blockchain against attack vectors (to maintain its reliability). But introducing new elements to a blockchain expose it to the possibility of infant mortality failure modes. So managing a blockchain's useful life is a complex tradeoff of risk and reliability concerns.

6. What is the meaning of reliability in the context of blockchain?

Any discussion about reliability requires defining the noun involved in the adjective of reliability: reliability of what. A blockchain is by nature reliable in that it is a distributed ledger, copied across multiple locations. Redundancy assures reliability against many failure modes, but not all. And the sheer scale of many blockchain networks assures it will almost never be fully functional, as at least one element may be in failure or disconnected from the network at a given time. Still, the reliability of the blockchain may be far less important than the reliability of the information it contains, or the authentication of the participants, or the reliable responses it gives to translation applications, for example. Large networks are almost always experiencing a failure, yet they reliably support the services and applications that rely on them. The Internet is always experiencing a failure, but services that ride the Internet are generally reliable enough to use.

If a part of a blockchain network is separated from another part, then the network elements can no longer spread information, which is a key function of a blockchain network. If parts of the network can't share information, then the blocks they create will be different, and the chain will split. It is reasonable to assume that eventually the network will rejoin, so that the separation is transient. Eventually, the rejoined blockchain network must prune one branch. For example, in bitcoin, the network eventually will have a branch that is longer, and the shorter will prune. But generally, the mechanism used to decide which is the valid branch in the chain must consider overall reliability and intent; the surviving part must have reliable information, and there should be a recovery mechanism so that reliable information from the other branch is not lost.

Therefore, we can talk about blockchain reliability from multiple frames of reference, and each of them has merit and importance. It is important to consider the reliability of the elements of the blockchain, the overall blockchain itself, the ledgers that are distributed on the blockchain, the information stored in those ledgers, and the applications that rely on it all.

Conclusion

Blockchain networks and the distributed ledgers they maintain have utility in the cable industry, when properly designed and applied. There remains a lot of hype and insufficient clarity around these technologies, so it is important to consider carefully what blockchain networks and distributed ledgers are good for, what they rely on or assume, and what additional work is needed to make them work.

While there will certainly be new emerging use cases in and outside the cable industry in the years to come, it is important to first consider the immediate opportunities. A useful categorization of use cases for operators includes new sources of revenue, ways to use these technologies to reduce costs, applications of these technologies to improve service or enhance the customer experience, and ways to reduce friction or simply make things easier to do.

But to take full advantage of blockchains and distributed ledger technologies, there are several design concerns to address carefully. Predominantly, we believe security and reliability issues are important at

this stage of the lifecycle of blockchain. Instead of avoiding complex problems, these technologies require us to solve some complex problems in security and reliability before we can truly benefit. Fortunately, while complex, the work is reasonable and doable.

Abbreviations

AR	augmented reality
BFT	Byzantine fault tolerance
CMTS	cable modem termination system
CPE	customer premise equipment
ICO	initial coin offering
IP	Internet protocol
NFV	network function virtualization
PKI	public key infrastructure
VoD	video on demand
VR	virtual reality

Bibliography & References

[1] “A Simple Overview of Blockchains, Why They Are Important to the Cable Industry.” Steve Goeringer. SCTE-ISBE. 2017

[2] “A Framework for Determining Blockchain Applicability.” Brian A. Scriber, Cablelabs. IEEE. 2018.

[3] “Mastering Bitcoin,” Andreas M. Antonopoulos, O’Reilly, 2010.

[4] “Sybil attack,” Wikipedia, downloaded 2018, https://en.wikipedia.org/wiki/Sybil_attack.

[5] “For Want of a Stronger Chain,” Jason Rupe, IEEE Blockchain Newsletter, July 2018, https://blockchain.ieee.org/newsletter/july-2018/for-want-of-a-stronger-chain.

[6] “The Byzantine Generals Problem”, Leslie Lamport, Robert Shostak, and Marshall Pease, July 1982, ACM Transactions on Programming Languages and Systems, Vol 4, No 3.

[7] “Practical Byzantine Fault Tolerance and Proactive Recovery”, Miguel Castro and Barbara Liskov, November 2002, ACM Transactions on Computer Systems, Vol. 20, No. 4.

[8] “The Part-Time Parliament”, Leslie Lamport, May 1998, ACM Transactions on Computer Systems.

[9] “The DAO, The Hack, The Soft Fork, and the Hard Fork”, CryptoCompare, Online, https://www.cryptocompare.com/coins/guides/the-dao-the-hack-the-soft-fork-and-the-hard-fork/, downloaded August 2018.

