A Model for Collaborative Blockchain-Based Video Delivery Relying on Advanced Network Services Chains

Nicolas Herbaut and Daniel Negru

The authors study how blockchain-powered smart contracts and network service chaining can be exploited to support such novel collaboration schemes. Their findings suggest that the proposed solution can complement existing technologies by supporting a wide range of business cases while at the same time significantly reducing costs.

ABSTRACT

The constant rise of over-the-top video consumption nowadays challenges the current Internet architecture. In this article, we propose a user-centric approach that helps the necessary reshaping of the content delivery ecosystem. We study how blockchain-powered smart contracts and network service chaining can be exploited to support such novel collaboration schemes. Finally, our findings suggest that the proposed solution can complement existing technologies by supporting a wide range of business cases while at the same time significantly reducing costs.

INTRODUCTION

In 2001, Marc Prensky coined the term "Digital Natives," describing what he perceived as a discontinuity in the education world due to the arrival and rapid dissemination of digital technology in a class of age.

Coming to adulthood, this generation is reshaping the TV industry by adopting over-the-top (OTT) services as their primary channel to consume a ubiquitous, on-demand, user-centric entertainment experience. Since the 2010s, this phenomenon has taken off so well that, according to Cisco VNI forecast [1], IP video will reach 82 percent of all IP traffic in 2020. Confronted with the challenges of delivering high-quality content to an ever growing number of users, a new type of architecture started to emerge. This layered delivery architecture promotes a clear separation between:

1. Hardware vendors
2. Content personalization systems
3. Content owners
4. Content providers (CPs)
5. Technical enablers (TEs)
6. Internet service providers (ISPs)
7. End users

In the near future, this model will be strongly challenged, given the current trends toward vertically integrated services. For example Netflix stopped using third party content delivery network (CDN) providers, relying exclusively on its own Open Connect system, making a single company responsible for recommending, selling, producing, owning, and delivering content [2].

In [3], Chuang advocates for future Internet architectures to be "designed for competition," as a means to achieve greater health and sustainability for the network.

The main factor toward ensuring such a design is to *permit different players to express their preferences for a service delivered by various providers*. Following this nomenclature, we identify, in Fig. 1, the six *loci* of competition of the content delivery market. Businesses often span over several competition loci, leaning toward more vertically integrated services. Controlling a certain locus has repercussions on others. For example, an end user cannot choose an alternative TE once he has chosen a CP. This article is focused on the most competition-challenged ones.

To represent the dynamics behind content delivery, Fig. 2 shows the functional interactions between stakeholders. First, the end user initiates a content query; then the CP, TE, and ISP collaborate to run a "content session" representing the actual consumption of the media by the end user. This schema highlights the current status quo in content delivery, but at the same time, it can also serve as the starting point of a more competitive ecosystem design, where:

- The end user expresses his desire to watch a specific content, along with quality of experience (QoE) specifications (e.g., minimal video resolution) in an enriched content query.
- Several CPs read the query and respond with a content offer.
- Several TEs offer their collaboration on the content delivery session, each relying on different technologies and network configurations.

The best content session should be dynamically negotiated between actors providing the desired QoE at the lowest cost.

The cost can be broken down into three parts:

1. The licensing cost charged by the CP to provide access to the content
2. The delivery cost charged by the TE for hosting and delivering the content
3. The network cost charged by the ISP to the TE to transfer the content to the end user

Implementing a trusted, scalable platform able to handle negotiation messages from different stakeholders and process them according to specific business rules can be highly challenging. In this context, the blockchain is perceived as an efficient novel software architecture building block

Digital Object Identifier: 10.1109/MCOM.2017.1700117

The authors are with the University of Bordeaux.

that allows reaching a distributed consensus for transactional data without the need for a trusted centralized party [4]. It consists of a read and append-only distributed database that maintains a list of records, called blocks, secured from tampering and revision as each block contains a timestamp and a link to the previous block. Blockchain offers the assurance that data cannot be modified retroactively once recorded. A decentralized consensus can be achieved using specific algorithms such as proof-of-work, proof-of-stake, or Practical Byzantine Fault Tolerance (PBFT). Blockchains can be used in a wide variety of use cases, such as monetary transactions like Bitcoin [5], medical records, and even network control [6].

This article proposes a model for collaborative blockchain-based video delivery. First, a decentralized brokering mechanism is introduced to create content sessions through the collaboration of a CP and a TE. Second, dynamic service chains are exploited in order to benefit from link diversity of different TEs, including user-centric resources.

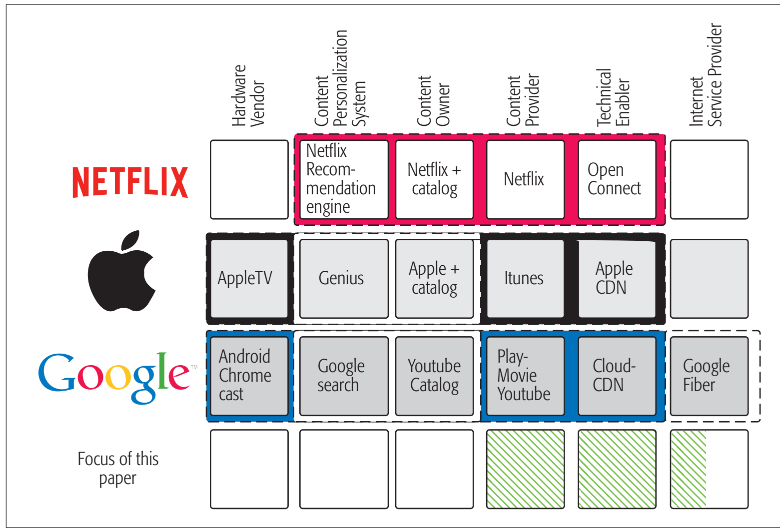

Figure 1. Competition loci in OTT content delivery.

A Model for Collaborative Video Delivery Based on Blockchain and Network Functions Virtualization Concepts

In this section, we describe a model using a blockchain to implement a decentralized brokering mechanism enabling a CP and a TE to compete and collaborate for the instantiation of the best content delivery session. After negotiation, this session is implemented as a network service chain the composition of which depends on the underlying technology of its network functions components — legacy CDN server, ISP virtual CDN-as-a-virtual network function (VNF), or user-centric VNF.

A Blockchain-Based Content Delivery Management Mechanism

From Blockchain to Smart Contracts: Current popular implementations of blockchains, such as the one supporting Bitcoin, have been successful at handling simple monetary transactions. However, the lack of native support for advanced programmability encouraged the development of a new generation of blockchain, extending the semantics of transaction through "smart contracts." Written in a Turing-complete language, smart contracts can process data on-chain to implement complex business rules. They can be useful in automating business processes in a trusted way, by allowing all stakeholders to process and validate contractual rules as a group [7].

Implementing Content Delivery Processes with Smart Contracts: We envision a content delivery brokering mechanism as a series of small smart contracts. Each contract has a unique identifier and some data fields, and can perform actions such as creating a new contract or updating the state of the blockchain. Contracts actions are triggered off by on-chain data update (i.e., creation of a new contract) or time.

The proposed model, as shown in Fig. 3, is composed of several blockchains, each one implementing a specific feature used for content distribution, as follows:

- The **content brokering blockchai**n handles the negotiation of the content delivery session. End users, CPs, and TEs publish smart contracts that will be used to determine the best mix for the session.
- The **delivery monitoring blockchain** collects and processes proofs of fulfillment of the delivery contract.
- The **provisioning blockchain** is used by CPs to handle the diffusion of contents on the TE's storage devices.

Content Delivery Session Brokering: Once the query arrives in the blockchain, a content brokering contract (CBC) is created (Fig. 3, step 1) and published. This contract specifies which content *c* to deliver and some user preferences, such as the expected target quality (e.g., 1080p). Then the CPs are notified of the new CBC contract, and use it to create content licensing contracts (CLCs) (step 2). The CLCs specify the price at which each CP is ready to sell content *c* to the end user, a reference to the CBC, and the maximum price for delivery. Next, once the CLCs are visible on the blockchain, TEs respond by publishing *content delivery contracts* (CDCs) (step 3), which specify the cost they are willing to charge for delivering content *c* to the user and the reference to the CLC. Finally, the original CBC collects all the related CDCs and arbitrates toward the cheapest one (step 4). All other contracts are terminated, and the winning contract is used to implement the content delivery. Relevant technical information required to implement the contract, such as content ID, TE ID, and end-user IP, are compiled in a content delivery service description (CDSD) document. We later detail how the contract is used to configure network service chains.

Content Delivery Session Monitoring: Smart contracts can implement a currency system to be used as a collateral means for ensuring the correct execution of the content delivery. Once the collaboration between each actor is formalized in a CDC, the payers (end user and CP) transfer their due payments to the CDC, which behaves as an escrow account. Each partner sends a proof of activity to the delivery monitoring blockchain according to its role in the content delivery. For

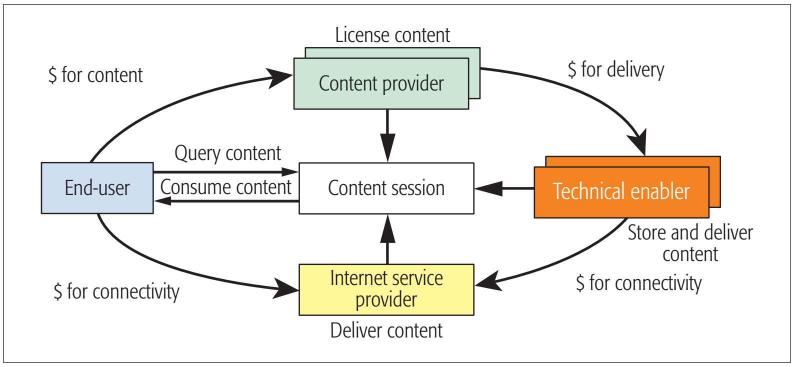

Figure 2. Stakeholders' interactions in the content session.

example, the CP could publish a cryptographic proof (e.g., as in the case of digital rights management) that entitles the TE to deliver the content. The end user publishes a proof of reachability of the content, whereas the TE publishes a proof of transmission. Once all the proofs are collected, the beneficiaries receive their payments. If the contract detects that a party has not fulfilled its duties, penalties can be applied, and some of the initial payment is refunded to payers.

Content Provisioning: Content dissemination throughout the network is key to reducing the price of delivery, and is usually achieved thanks to a resource prediction engine [8]. The blockchain can be used for content provisioning in TEs in two ways. The TEs can audit the content popularity from chain data and proactively decide to *pull* contents from the CP to subsequently sell CDCs. Alternatively, the CPs can *push* contents to the TEs by rewarding them through the blockchain in compensation for storage. CPs benefit from having their content widespread on the network, since more TEs publishing a CDC means more competition and lower price.

Governance Models

As the proposal relies on a fully decentralized agreement conclusion mechanism, we need a way to establish the respective liabilities of stakeholders in case of problems. As smart contracts are not legal contracts in essence, any litigation should be solved by proper prior legal agreements. Several models can be considered:

- Chain of responsibility: Each actor contracts with a supplier, which is liable for the service it provides. CPs are liable toward end users, TEs are liable toward CPs, and TEs are liable toward ISPs. This solution is not very scalable as it implies having thousands of contracts.
- Consortium: Actors create a consortium providing the legal foundations for the service. The consortium manages any liabilities centrally and automatically thanks to the blockchain. This model opposes the decentralization of transactions, but offers a more scalable alternative.
- Decentralized autonomous organization: In this model, legal aspects are directly managed on-chain by an organization the governance of which is defined by the code of smart contracts, bringing full decentralization and automation. However, the legal status of this type of business organization is still unclear.

Our proposal fosters competition by allowing several actors to offer their resources to the system and adjust their prices to match demand. By decoupling the content delivery from the content licensing, we set up a much more diverse ecosystem, by including actual end users (assuming the role of TEs) in the content delivery process. However, constructing content sessions by using third party resources induces a challenge to current Internet architectures. In the next section, we describe how content sessions can be dynamically mapped to network service chains through network softwarization and the use of microservices.

Instantiating the Model through Advanced Dynamic Network Service Chains

Once the brokering of content licensing and delivery is complete, the content session between the TE and the end user is implemented. Content sessions are on-demand, user-centric service chains deployed based on the specifications of the CDSD. The deployment of the service chain is shared between ISPs and TEs, the ISPs being responsible for steering the traffic of the end user to/from the TE domain, while the TEs implement both networking and service configuration of IP endpoints.

TEs implement content delivery in several ways. We detail three complementary approaches, CDN, vCDN, and µCDN, detailed below. Figure 4 shows the deployment of these three types of TE. User 1 gets its content from a CDN, while User 2 uses a vCDN service chain deployed in the ISP network. User 3 retrieves its content directly from user 4's µCDN.

CDN Delivery: This is the classical case used today in OTT scenarios where the content is hosted on servers belonging to another autonomous system (AS) with no possible end-to-end management. The content is delivered in best effort mode from the server, selected by the CDN operator depending on the end user's physical location through dynamic Domain Name Service (DNS) resolution. Little to no collaboration occurs in this scenario, and the server selection made by the CDN operator may not be in line with the traffic engineering objectives implemented by the ISP. To circumvent this issue, the next two sections present the deployment of network service chains where the network is handled by the ISP end to end, allowing the implementation of quality of service policies.

vCDN Delivery: This solution aims at instantiating a CDN as a VNF inside the ISP AS, deployed on a network functions virtualization (NFV) infrastructure point of presence (NFVI-POP). The vCDN can be operated by the ISP [9] or by an external CDN operator leasing the ISP infrastructure. This approach reduces the hop count between client and server, and supports end-to-end management of the service through a service level agreement (e.g., imposing minimal bandwidth or maximal delay). Our previous work [10] described a network service chain that can be used to handle both routing (with the virtual media gateway — VMG) and content distribution (with the Virtual Streamer — vStre). Deploying a service over an NFV infrastructure over multiple data centers [11] usually relies on software defined networking (SDN) for flexibility.

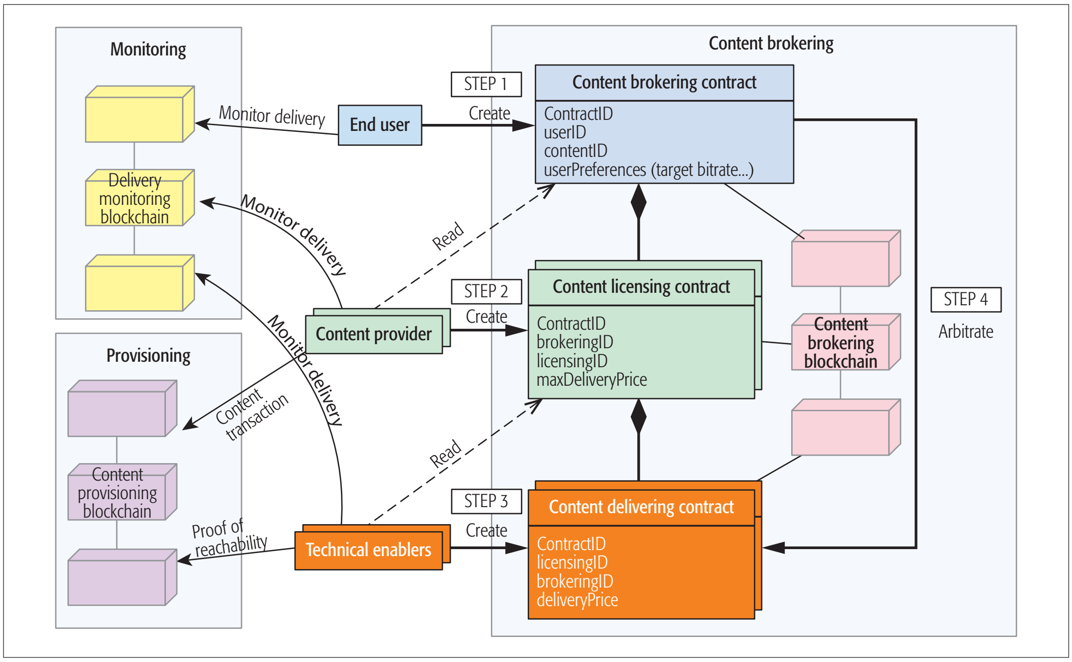

Figure 3. Blockchain-based model for collaborative video delivery.

The Blockchain can be used for content provisioning in TEs in two ways. The TEs can audit the content popularity from chain data and pro-actively decide to pull contents from the CP to subsequently sell CDCs. Alternatively, the CPs can push contents to the TE by rewarding them through the Blockchain in compensation for storage.

µCDN Delivery: Customer premises equipments (CPE) provides plenty of spare system and network resources that can be used for content delivery. With modern operating systems (OSs, e.g., GNU/Linux, Android) they can support the deployment of new services and even VNFs [12]. Their small scale, however, requires downscaling the main concepts behind NFV.

Figure 4 shows the internal microservice architecture used to implement the µCDN. The two key technologies we use to address the above-mentioned challenges are containers and SDN, as follows:

- Containers are lightweight virtualization mechanisms that bundle applications and their dependencies. With their reduced footprint and low CPU overhead, they are often considered in cloud edge architectures [13].
- The SDN-capable software switch deployed on the CPE allows manipulating the containers' connectivity in an OS-independent fashion thanks to the use of standard protocols such as OpenFlow.

Service deployment is triggered by the publication of the CDSD on the blockchain (Fig. 4, 1). It is retrieved by the µOrchestrator module, which spawns content delivery containers running HTTP servers able to stream the content (2a) and configures their network (2b). A CP may choose to use different technologies to license their content, from simple files to more complex DRM-based solutions. Adopting a microservice architecture, our solution keeps these implementation details in the content delivery container and ensures that, regardless of the underlying technology, the CDSD provides all resources needed (e.g., cryptography material). Finally (3a), the µOrchestrator instructs the SDN controller to update the network configuration, which is, in turn (3b), deployed by the software switch so that the connection between the end user and the content delivery container can be established.

Evaluation of the Proposed Model

Network Services Chain Evaluation

We implemented a discrete event simulator with the SimPy Library to emulate content delivery sessions. We simulated 15,000 content session requests spanning over 25 minutes. For every request, each TE that:

1. Stores the content
2. Has enough bandwidth to deliver the content

asks for a delivery price assumed to be proportional to the number of hops between itself and the end user.

The brokered price corresponds to the smallest price demanded by a TE. CDNs were assumed to host the entire content catalog, whereas µCDN and vCDN pulled the content from the CP by auditing blockchain data and downloading the most popular contents. We used a real ISP topology of 2k nodes and 60k edges extracted from the Center for Applied Internet Data Analysis. Six CDNs were placed in a weighted random fashion at the most connected links, which correspond to the Internet exchange points on the operator topology. We then placed 500 service access point nodes representing the user location in the network in a similar way, selecting the least connected links. Finally, 100 vCDNs and 500 µCDNs were randomly distributed among the nodes with connectivity degrees in the middle range (40–90 percent). vCDNs' capabilities were based on common virtual caching appliances' specifications (1 TB of storage supporting 150 Mb/s or 30 concurrent 720p streaming sessions), while µCDN capabilities were based on current CPE specifications (30 GB of storage, 20 Mb/s of upload speed, or 4 concurrent 720p streaming sessions). Contents stored in µCDNs and vCDNs are purged according to a least recently used rule. CDNs were assumed to support a large amount of concurrent connections (2.5 Gb/s or 500 sessions). We assumed content popularity to follow a

The need for performance and Smart Contracts support compelled us to use the open source project Hyperledger-Fabric (www.hyperledger.org/projects/fabric). This Linux foundation project can be used to build Blockchain solutions with a modular architecture to deliver flexibility and scalability.

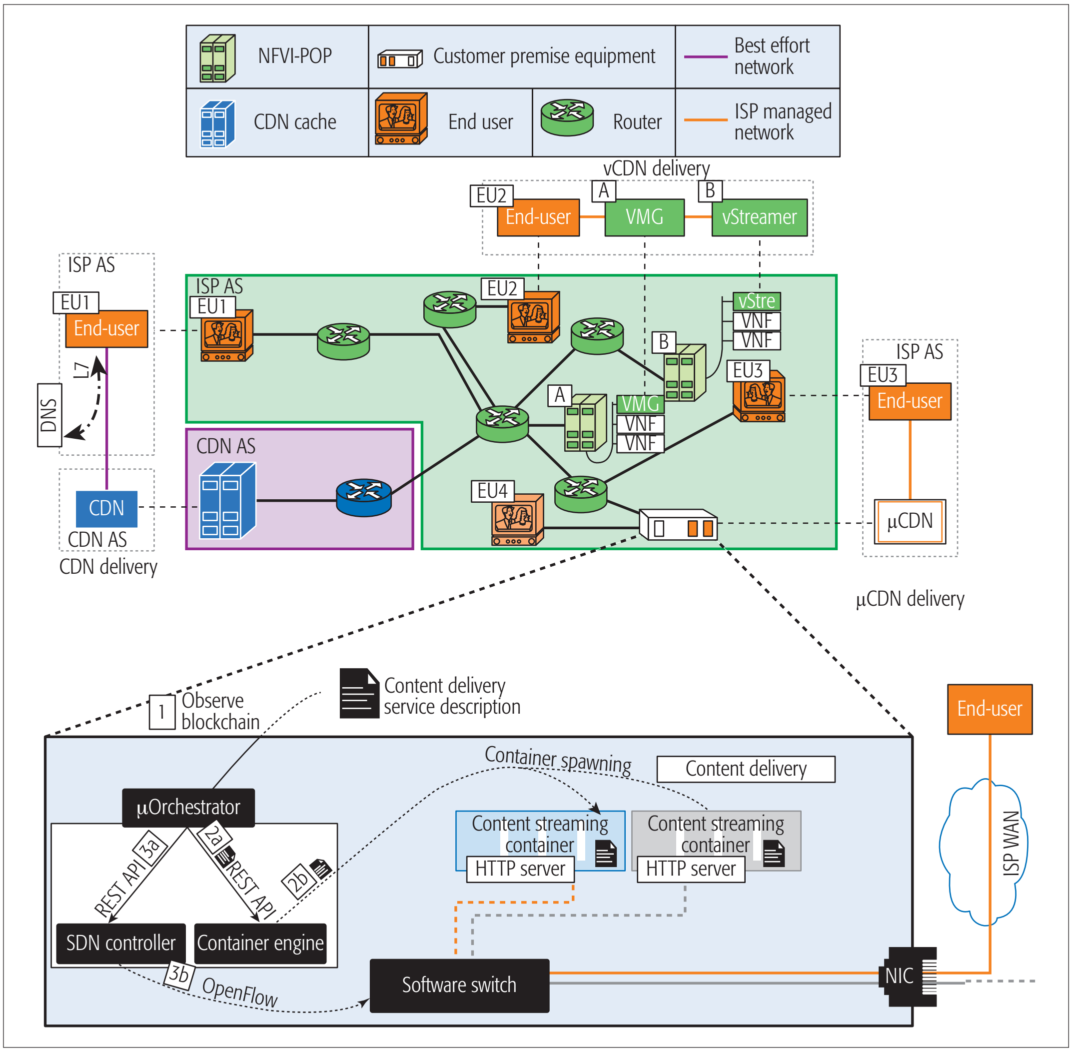

Figure 4. CDN, vCDN, and µCDN services deployed in an ISP network.

Zipf distribution. The hop count is computed from the topology for vCDNs and µCDNs; however, for the CDN, we assume that three additional hops are used within the CDN network between the edge of the ISP network and the final server, corresponding to the average ISP graph distance.

Results of the experiments are presented in Fig. 5. Figure 5a shows the respective shares of TEs. At the beginning we see that every request is served by the CDNs, as they are still the only ones hosting the content. After 2 min, once the popular contents are downloaded by the vCDNs, they also start delivering contents. The reason vCDNs are privileged w.r.t. CDNs is that they are spread more widely in the network, with a smaller average distance to end users. After 3 min, the µCDNs start serving content as well, and their share increases to 12 min, where they become the most used TEs. Again, this can be explained by a denser distribution of µCDNs in the network causing a lower hop count. After the 20 min mark, the shares stabilize. Despite their advantage, µCDNs only absorb half of the content requests. In fact, due to their limited capacity and storage, they are able to store and deliver only very popular contents. vCDNs store both very popular contents and less popular contents and still account for a third of content sessions. Finally, CDNs absorb the long tail of contents that are not popular enough to be stored by other TEs.

Another important benefit of our solution is the hop count reduction. Figure 5b compares the average number of hops between the selected TE and the end user when using all three TE types in conjunction, but also using only some of them. We can see from the figure that using only the CDNs yields a higher hop count, stable over time. When complementing a CDN with vCDNs, the hop count sharply decreases, as content gets stored near the edges of the network, and stabilizes near the four-hop mark. When using both CDN and µCDN, the curve decreases slowly, as contents take more time to be provisioned at the edges. Finally, using all three TEs yields the lowest hop count, with a fast drop at the beginning and a downward trend reaching the lowest value of our experiment.

Blockchain Evaluation

Test Environment: Our goal is to build a system where each content session is brokered on the blockchain. For this reason, its *performance*, measured in terms of number of transactions processed per second, is key to providing the content sessions quickly. At the same time, we envision the number of "clients" (end users, CPs, and TEs)

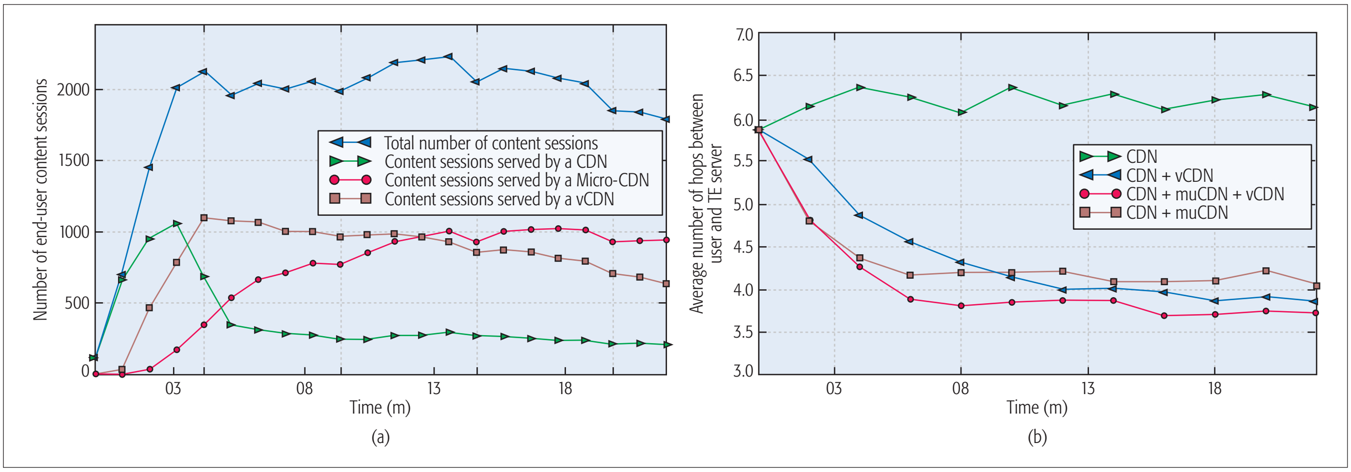

Figure 5. Network services chain evaluation: a) respective TEs share for CDC; b) average price for content delivery.

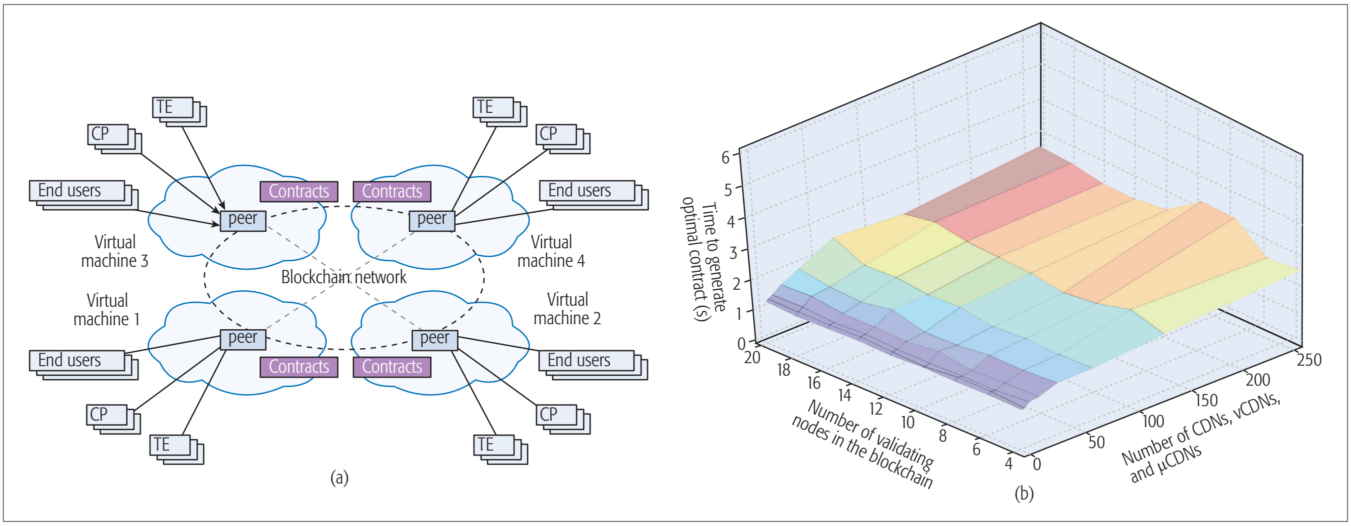

Figure 6. Blockchain evaluation: a) testbed; b) performance and scalability experiment.

using the service to be high, so the blockchain must ensure good *node scalability*. Today, "permissionless" blockchains based on proof-of-work consensus offer great node scalability, but lack the required throughput (e.g., up to 7 tx/s with Bitcoin). On the other hand, blockchains based on advanced Byzantine Fault-Tolerant (BTF) state-machine replication protocols offer excellent performance in terms of throughput and latency but require all nodes to know the IDs of all other nodes [14]. In our case, we used a "permissioned" blockchain as the nodes processing the transactions do not need to be anonymous.

The need for performance and smart contract support compelled us to use the open source project Hyperledger-Fabric (www.hyperledger.org/projects/fabric). This Linux foundation project can be used to build blockchain solutions with a modular architecture to deliver flexibility and scalability. It provides pluggable consensus algorithms (by default PBFT) and simple smart contract implementation in Go or Java.

The critical aspects of the brokering mechanism is the time needed to converge toward the optimal CDC, involving the end user, the CP, and the TE. This delay affects the end-user QoE, as the content delivery session can start only after the optimal CDC is computed. A lot of contracts are published in the blockchain; for example, if we assume that there are 10 CPs and 100 TEs, up to 10 × 100 contracts will be published.

Considering this, the evaluation is focused on the content brokering blockchain as it is the most time-sensitive and subject to scalability issues.

We deployed the solution with Hyperledger-Fabric configured with the PBFT consensus, as shown in Fig. 6a. We then paired end-user applications (publishing CBC), CP applications (reading CBC from the blockchain and responding by publishing CLCs) and TE applications (reading CLCs and publishing CDCs), and the blockchain validating peers, which are the nodes responsible for running the consensus, validating transactions, and maintaining the ledger.

Each user was configured to send 10 requests/min. We then computed the average time needed to obtain the optimal CDC, or convergence time. We varied the number of TE agents, with the number of CPs being fixed at 10.

The results presented in Fig. 6b show that for 50 TEs, the convergence time is below 2 s. This time increases for higher values of TEs, reaching 4 s in the worst case scenario of 250 TEs, which remains acceptable.

On our testbed, the number of nodes increased the convergence time slightly. This is due to the rather good networking performance of our cloud instances, located in the same availability zone. In a production deployment, nodes would not be collocated to improve resiliency, and the performance might be even more impacted.

Discussion on Scalability: The blockchain network is composed of validating nodes that run the smart contracts and append blocks to the chain once consensus is reached. They are also used to query the state of the blockchain by clients. Increasing the number of validating nodes has two antagonistic effects:

1. Each node serves fewer clients, reducing the average number of requests per node.
2. The quorum needed for the consensus is increased, increasing the number of messages shared in the network.

On our testbed, the number of nodes increased the convergence time slightly. This is due to the rather good networking performance of our cloud instances, located in the same availability zone. In a production deployment, nodes would not be collocated to improve resiliency, and the performance might be even more impacted.

The next release of Hyperledger-Fabric and recent research papers such as [15] promote new architectures that support parallelizing the validation of transactions through their endorsement by only a subset of nodes. In this perspective, transactions are managed on sub-chains supporting fine-tuned consensus algorithms, improving scalability.

Conclusions and Future Work

This article proposes a new model for content distribution over the Internet, with a scalable blockchain-based brokering mechanism allowing several providers to collaborate and to provide the requested service through network service chains. On top of reducing the overall delivery cost, it promotes healthy competition, allowing user-centric resources to join the game and paving the way for new business models.

Further work will consist of evaluating the scalability of the solution on a production-ready blockchain architecture. The issues of governance, security, and privacy for end users have not been addressed and need further investigation.

Acknowledgments

We thank Marko Vukolić (IBM Research-Zurich), who provided valuable feedback at early stages of this article. The work performed for this article has been partially funded by the FP7 IP T-NOVA European Project (Grant Agreement N#619520) and the FUI French National Project DVD2C.

References

[1] Cisco Visual Networking Index, Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015–2020 white paper.

[2] T. Böttger *et al.*, "Open Connect Everywhere: A Glimpse at the Internet Ecosystem Through the Lens of the Netflix Cdn," arXiv preprint arXiv:1606.05519, 2016.

[3] J. Chuang, "Loci of Competition for Future Internet Architectures," *IEEE Commun. Mag.*, vol. 49, no. 7, July 2011, pp. 38–43.

[4] X. Xu *et al.*, "The Blockchain as a Software Connector," *2016 13th Working IEEE/IFIP Conf. Software Architecture*, 2016, pp. 182–91.

[5] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[6] N. Bozic, G. Pujolle, and S. Secci, "A Tutorial on Blockchain and Applications to Secure Network Control-Planes," *Smart Cloud Networks & Systems*, 2016, pp. 1–8.

[7] R. Hull *et al.*, "Towards a Shared Ledger Business Collaboration Language Based on Data-Aware Processes," *Int'l. Conf. Service-Oriented Computing*, Springer, 2016, pp. 18–36.

[8] Y. Kryftis *et al.*, "Efficient Entertainment Services Provision over a Novel Network Architecture," *IEEE Wireless Commun.*, vol. 23, no. 1, Feb. 2016, pp. 14–21.

[9] P. A Frangoudis *et al.*, "An Architecture for On-Demand Service Deployment over a Telco CDN," *IEEE ICC*, 2016, pp. 1–6.

[10] N. Herbaut *et al.*, "Service Chain Modeling and Embedding for NFV-Based Content Delivery," *IEEE ICC*, 2017.

[11] R. Mijumbi *et al.*, "Network Function Virtualization: State-of-the-Art and Research Challenges," *IEEE Commun. Surveys & Tutorials*, vol. 18, no. 1, 2016, pp. 236–62.

[12] D. Minodier and G. Dalle, "Juniper, Network Enhanced Residential Gateway," tech. rep. TR-317, Broadband Forum, July 2016; https://www.broadband-forum.org/technical/download/TR-317.pdf.

[13] C. Pahl and B. Lee, "Containers and Clusters for Edge Cloud Architectures — A Technology Review," *2015 3rd Int'l. Conf. Future Internet of Things and Cloud*, 2015, pp. 379–86.

[14] M. Vukolic, "The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication," *Int'l. Wksp. Open Problems in Network Security*, Springer, 2015, pp. 112–25.

[15] W. Li *et al.*, "Towards Scalable and Private Industrial Blockchains," *Proc. ACM Wksp. Blockchain, Cryptocurrencies and Contracts*, 2017, pp. 9–14.

Biographies

Nicolas Herbaut (nicolas.herbaut@labri.fr) is a Ph.D. candidate at Bordeaux University doing research on virtualization, network softwarization, and content delivery. He received an M.Sc.Eng. in mathematics from INSA Rouen in 2004, and an M.Econ. in economic analysis from Aix-Marseille University in 2005.

Daniel Negru (daniel.negru@labri.fr) is an associate professor at Bordeaux University, specializing in multimedia and networking. His current activities are focused on video streaming, content delivery, and NFV/SDN. He has participated in numerous collaborative research projects at the European level, and published more than 60 papers in journals and conferences such as *IEEE Communications Magazine*, *IEEE Multimedia*, ICME, GLOBECOM, and ICC.