HAL
open science

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

Nicolas Herbaut, Daniel Negru

To cite this version:

Nicolas Herbaut, Daniel Negru. A Model for Collaborative Blockchain-Based Video Delivery Relying on Advanced Network Services Chains. IEEE Communications Magazine, 2017, 55 (9), pp.70 - 76. 10.1109/MCOM.2017.1700117 . hal-01610670

HAL Id: hal-01610670

https://hal.science/hal-01610670v1

Submitted on 1 Jun 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Model For Collaborative Blockchain-based Video Delivery Relying On Advanced Network Services Chains

Nicolas Herbaut, Daniel Négru*

Abstract

The constant rise of Over-The-Top video consumption nowadays challenges the current Internet architecture. In this paper, 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 area of business cases while, at the same time, importantly reducing costs.

1 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 services as their primary channel to consume an 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% of all IP traffic in 2020.

Confronted to 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 (CP), (5) Technical Enablers (TE), (6) Internet Service Providers (ISP) and (7) End-Users. In the near future, such model will be strongly challenged, given the current trends towards vertically integrated services. For example Netflix stopped using third party Content Delivery Netrwork (CDN) providers, relying exclusively on its own Open Connect system, making a single company responsible for recommending, selling, producing, owning and delivering the content [2].

In [3], Chuang advocates for future Internet architectures to be "designed for competition", as a mean to achieve greater health and sustainability for the network. The main factor toward assuring such a design is to *permit different players to express their preferences for a service delivered by various providers*. Following his nomenclature, we identified, in Figure 1, the 6 *loci* of competition of the content delivery market. Company business often span over several competition loci, leaning toward more vertically integrated services. Controlling a certain loci has repercussion on others. For example, an End-User cannot choose an alternative Technical Enabler once he has chosen the Content Provider. The paper is focused on the most competition-challenged ones.

To represent the dynamics behind content delivery, Figure 2 shows the functional interactions between stakeholders. First, the End-User initiates a content query, then 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 specifications (e.g., minimal video resolution) in an enriched content query;
- *Several* Several Content Providers read the query and respond with a content offer;
- *Several* Several Technical Enablers offer their collaboration to the content delivery session, each relying on different technologies and network configurations.

*Nicolas Herbaut and Daniel Négru are with the University of Bordeaux, France

The best content session should be dynamically negotiated between actors providing the desired Quality of Experience with the lowest cost. The cost can be broken down into 3 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 and (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 message 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 that allows reaching a distributed consensus for transactional data without the need of a trusted centralized party [4]. It consists in 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 PBFT (Practical Byzantine Fault Tolerance) algorithm. Blockchains can be used in a wide variety of use cases, such as monetary transaction like Bitcoin [5], medical records or even network control [6].

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

2 A model for collaborative video delivery based on Blockchain and Network Function Virtualization concepts

In this section, we describe a model uising a Blockchain to implement a decentralized brokering mechanism enabling CP and 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 which composition 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).

2.1 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 native support for advanced programmability encouraged the development of a new generation of Blockchain, extending the semantic 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, 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 Figure 3, is composed of several Blockchains, each one implementing a specific feature used for content distribution, as follows:

- The **Content Brokering Blockchain** 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 process proofs of the fulfillment of the delivery contract;
- The **Provisioning Blockchain** is used by CPs to handle the diffusion of contents on the TEs storage devices.

2.1.1 Content Delivery Session Brokering

Once the query arrives in the Blockchain, a **Content Brokering Contract** (CBC) is created (Figure 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** (CLC) (step 2). The CLCs specifie the price at which each CP is ready to sell content c to the End-User, a reference of the CBC, and the maximum price for the delivery. Next, once the CLCs are visible on the Blockchain, Technical Enablers respond by publishing **Content Delivering Contracts** (CDC) (step 3) which specify the cost they are willing to charge for delivering content c to the user and the reference of the CLC. Finally, the original CBC collects all the related CDCs and arbitrates towards the cheapest one (step 4). All other contracts are terminated and the winning contract is used to implement the content delivery. Relevant technical informations 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. Section 2.3 details how the contract is used to configure network service chains.

2.1.2 Content Delivery Session Monitoring

Smart Contracts can implement a currency system to be used as a collateral mean 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 example, the CP could publish a cryptographic proof (e.g., case of a Digital Right Management) that it entitled 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.

2.1.3 Content Provisioning

Content dissemination throughout the network is key to reduce the price of the 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 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. CPs benefit from having their content widespread on the network, since more TEs publishing CDC means more competition and lower price.

2.2 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 and liable toward ISPs. This solution is not very scalable as it would 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 offer a more scalable alternative.
- Decentralized autonomous Organization: in this model, legal aspects are directly managed on-chain by an organization which governance 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 adjusting 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 TE) 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.

2.3 Instantiating the model through advanced dynamic network services chains

Once the brokering of content licensing and delivery is complete, the Content Sessions 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 thereafter. 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.

2.3.1 CDN Delivery

It 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 physical location through dynamic 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.

2.3.2 vCDN Delivery

This solution aims at instantiating a CDN as a Virtual Network Function inside the ISP AS, deployed on an NFV Infrastructure Points-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 a NFV infrastructure over multiple datacenters [11] usually relies on Software Defined Networking (SDN) approach for flexibility.

2.3.3 μCDN Delivery

Customer Premise Equipments (CPE) provides plenty of spare system and network resources that can be used for content delivery. With modern operating systems (GNU/Linux, Android) they can support the deployment of new services and even Virtual Network Functions [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). Content Provider may choose to use different technologies to license their content, from simple file to more complex DRM-based solutions. Adopting a microservice architecture, our solution keeps these implementation details in the Content Delivery Container and assures that, regardless of the underlying technology, the Content Delivery Service Description provides every resources needed (e.g., cryptography material). Finally, (3a) the μOrchestrator informs SDN Controller to update the network configuration that 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.

3 Evaluation of the proposed model

3.1 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 requests, each TE that (1) stores the content and (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 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 Points 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%) vCDNs capabilities were based on common virtual caching appliances specifications (1TB of storage supporting 150 Mbps or 30 concurrent 720p streaming sessions) while μCDN capabilities were based on current customer premises equipment specifications (30GB of Storage, 20 Mbps of upload speed or 4 concurrent 720p streaming sessions). Contents stored in μCDN and vCDN are purged according to a Least Recently Used rule. CDNs were assumed to support a large amount of concurrent connections (2.5Gbps or 500 sessions). We assumed content popularity to follow a zipf distribution. The hop count is computed from the topology for vCDNs and μCDNs, however, for the CDN, we assume that 3 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 on Figure 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 mins, once the popular contents are downloaded by the vCDNs, they also start delivering contents. The reason why vCDNs are priviledged wrt CDNs is that they are spread more widely in the network, with a smaller average distance to End-Users. After 3 mins, the μCDNs start serving content as well, and their share increase up to 12 mins, 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 minutes 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 content and less popular contents and still account for a third of content session. Finally, CDNs absorb the long tail of contents, which 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 to a higher hop count, stable over time. When complementing CDN with vCDNs, the hop count sharply decreases, as content gets stored near the edges of the network, and stabilizes near the 4 hops mark. When using both CDN and μCDN, the curve decreases slowly, as contents

take more time to be provisioned in the edges. Finally, using all 3 TEs yields to the lowest hop count, with a fast drop at the beginning and a downward trend reaching the lowest value of our experiment.

3.2 Blockchain evaluation

3.2.1 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 transaction processed per second, is key to provide the content sessions quickly. At the same time, we envision the number of "clients" (End-Users, CP and TEs) using the service to be high, so the Blockchain must ensure a 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/second with Bitcoin). On the other hand, Blockchains based on advanced Bysantine 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 "permissionned" Blockchain as the nodes processing the transactions do not need to be anonymous.

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. 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 Content Delivery Contract, 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 CP and 100 TEs, then 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 on Figure 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), 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 per minute. We then computed the average time needed to obtain the optimal CDC, or convergence time. We varied the number of TEs agents, the number of CP being fixed at 10.

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

3.2.2 Discussion on Scalability

The Blockchain network is composed of validating nodes that run the Smart Contracts and append blocks to the chain once the consensus is reached. They are also used to query the state of the Blockchain by clients. Increasing the number of validating nodes has two antagonist effects: (1) each node serve less clients, reducing the average number of requests per node and (2) the quorum needed for the consensus is increased, increasing the number of message shared in the network.

On our testbed, the number of nodes increased slightly the convergence time. This is due to the rather good networking performances 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 may be even more impacted.

The next release of Hyperledger-Fabric and recent research papers such as [15] promotes new architectures that support parallelizing the validation of transaction 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.

4 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 a healthy competition allowing user-centric resources to join the game and paving the way for new business models.

Further work will consist in evaluating the scalability of the solution on a production ready Blockchain architecture. The issues of Governance, security and privacy for end-user has not be addressed need further investigation.

Acknowledgements

We thank Marko Vukolić (IBM Research - Zurich), who provided valuable feedback at early stages of this article. The work performed for this paper 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] Timm Böttger, Felix Cuadrado, Gareth Tyson, Ignacio Castro, and Steve Uhlig. Open connect everywhere: A glimpse at the internet ecosystem through the lens of the netflix cdn. *arXiv preprint arXiv:1606.05519*, 2016.

[3] John Chuang. Loci of competition for future internet architectures. *IEEE Communications Magazine*, 49(7):38–43, 2011.

[4] Xiwei Xu, Cesare Pautasso, Liming Zhu, Vincent Gramoli, Alexander Ponomarev, An Binh Tran, and Shiping Chen. The blockchain as a software connector. In *Software Architecture (WICSA), 2016 13th Working IEEE/IFIP Conference on*, pages 182–191. IEEE, 2016.

[5] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[6] Nikola Bozic, Guy Pujolle, and Stefano Secci. A tutorial on blockchain and applications to secure network control-planes. In *Smart Cloud Networks & Systems (SCNS)*, pages 1–8. IEEE, 2016.

[7] Richard Hull, Vishal S Batra, Yi-Min Chen, Alin Deutsch, Fenno F Terry Heath III, and Victor Vianu. Towards a shared ledger business collaboration language based on data-aware processes. In *International Conference on Service-Oriented Computing*, pages 18–36. Springer, 2016.

[8] Yiannos Kryftis, George Mastorakis, Constandinos X Mavromoustakis, Jordi Mongay Batalla, Evangelos Pallis, and Georgios Kormentzas. Efficient entertainment services provision over a novel network architecture. *IEEE Wireless Communications*, 23(1):14–21, 2016.

[9] Pantelis A Frangoudis, Louiza Yala, Adlen Ksentini, and Tarik Taleb. An architecture for on-demand service deployment over a telco cdn. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–6. IEEE, 2016.

[10] Nicolas Herbaut, Daniel Négru, David Dietrich, and Panagiotis Papadimitriou. Service chain modeling and embedding for nfv-based content delivery. In *IEEE International Conference on Communications (ICC)*, 05/2017 2017.

[11] Rashid Mijumbi, Joan Serrat, Juan-Luis Gorricho, Niels Bouten, Filip De Turck, and Raouf Boutaba. Network function virtualization: State-of-the-art and research challenges. *IEEE Communications Surveys & Tutorials*, 18(1):236–262, 2016.

[12] David Minodier, and Gregory Dalle, Juniper. Network Enhanced Residential Gateway. Technical Report TR - 317, Broadband Forum, July 2016. URL `https://www.broadband-forum.org/technical/download/TR-317.pdf`.

[13] Claus Pahl and Brian Lee. Containers and clusters for edge cloud architectures–a technology review. In *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*, pages 379–386. IEEE, 2015.

[14] Marko Vukolić. The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. In *International Workshop on Open Problems in Network Security*, pages 112–125. Springer, 2015.

[15] Wenting Li, Alessandro Sforzin, Sergey Fedorov, and Ghassan O Karame. Towards scalable and private industrial blockchains. In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, pages 9–14. ACM, 2017.

Nicolas Herbaut

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 a M.Econ in Economical Analysis from Aix-Marseille University in 2005. Contact him at nicolas.herbaut@labri.fr.

Daniel Negru

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

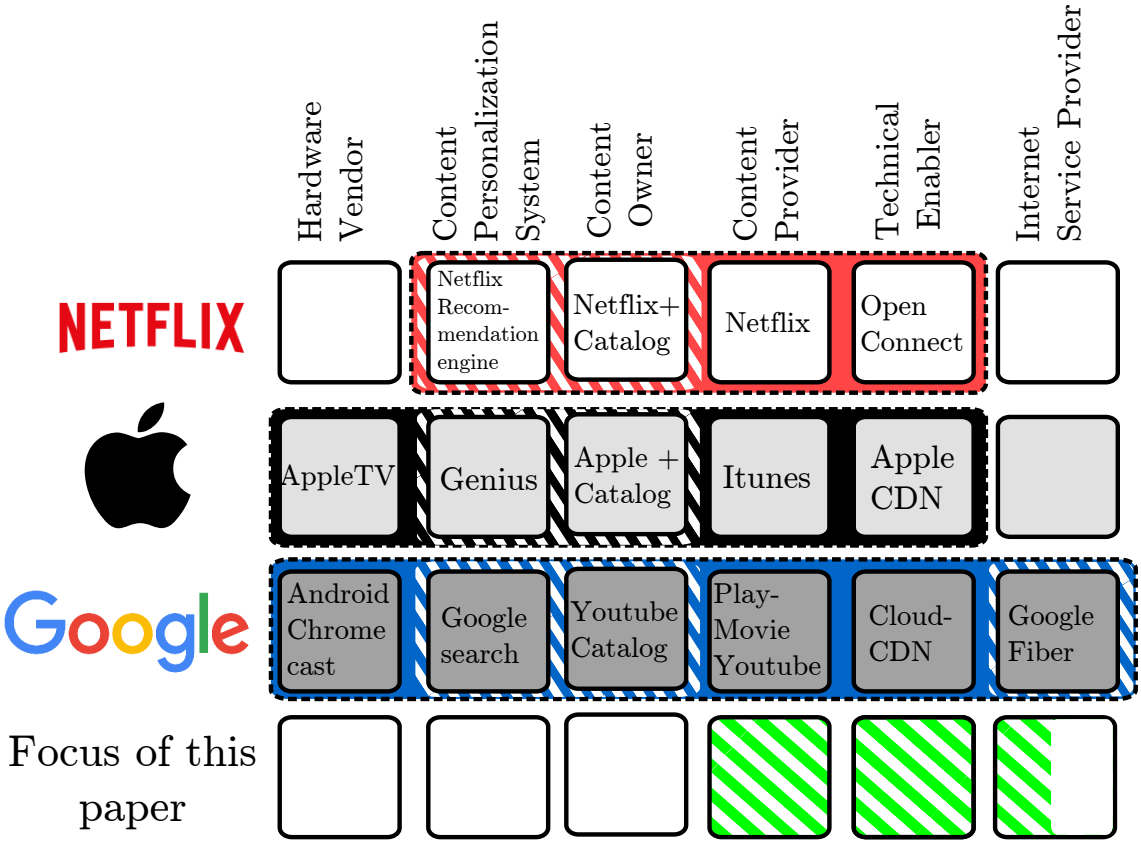


Figure 1: Competition Loci in OTT content delivery.

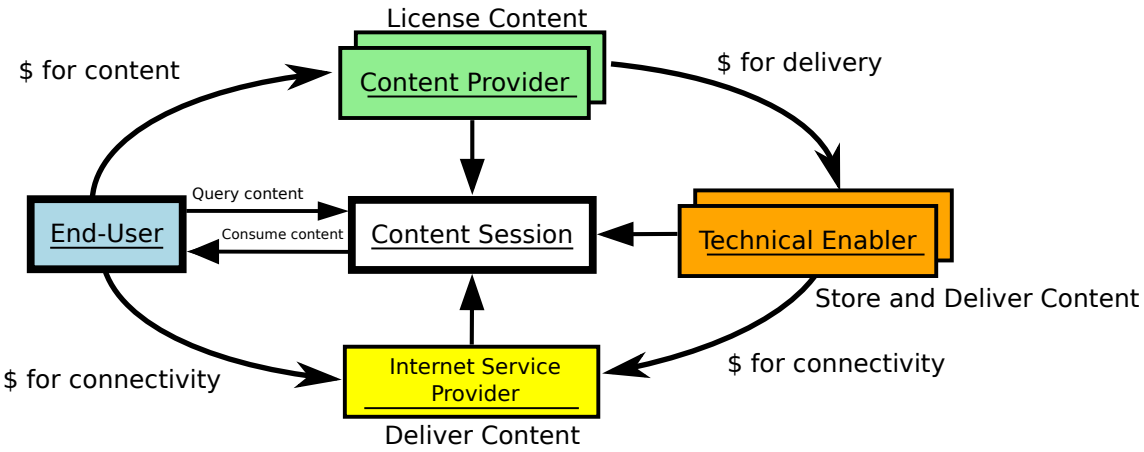


Figure 2: Stakeholders interactions in the content session.

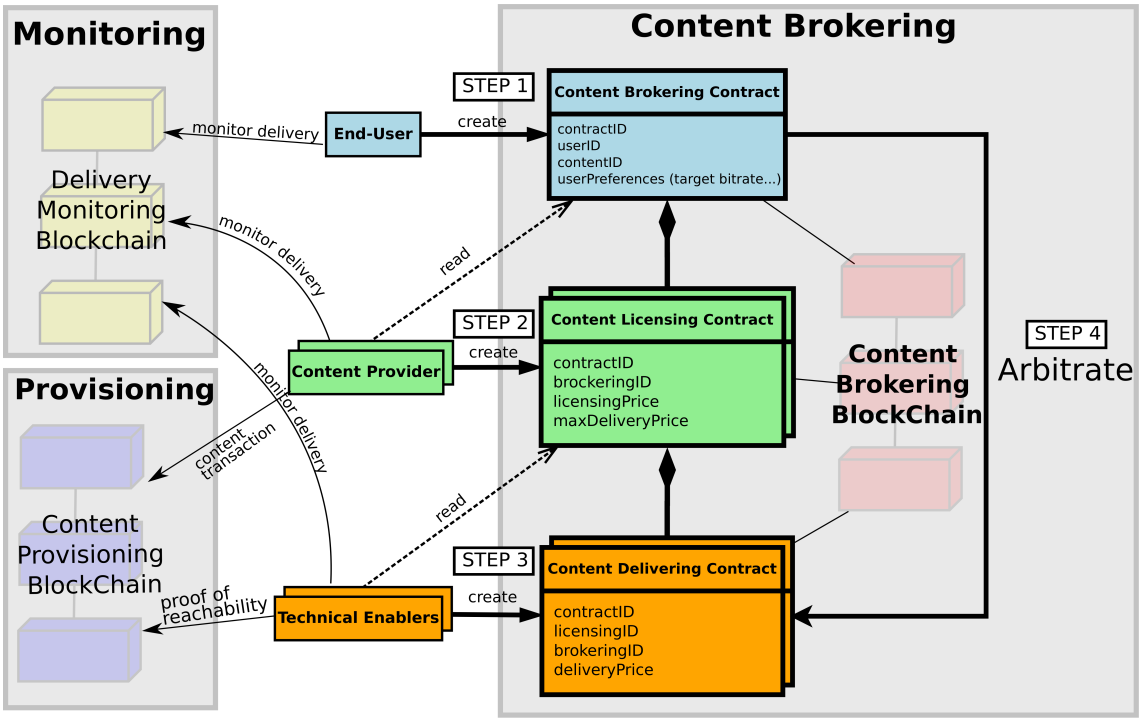


Figure 3: Blockchain-based model for collaborative video delivery.

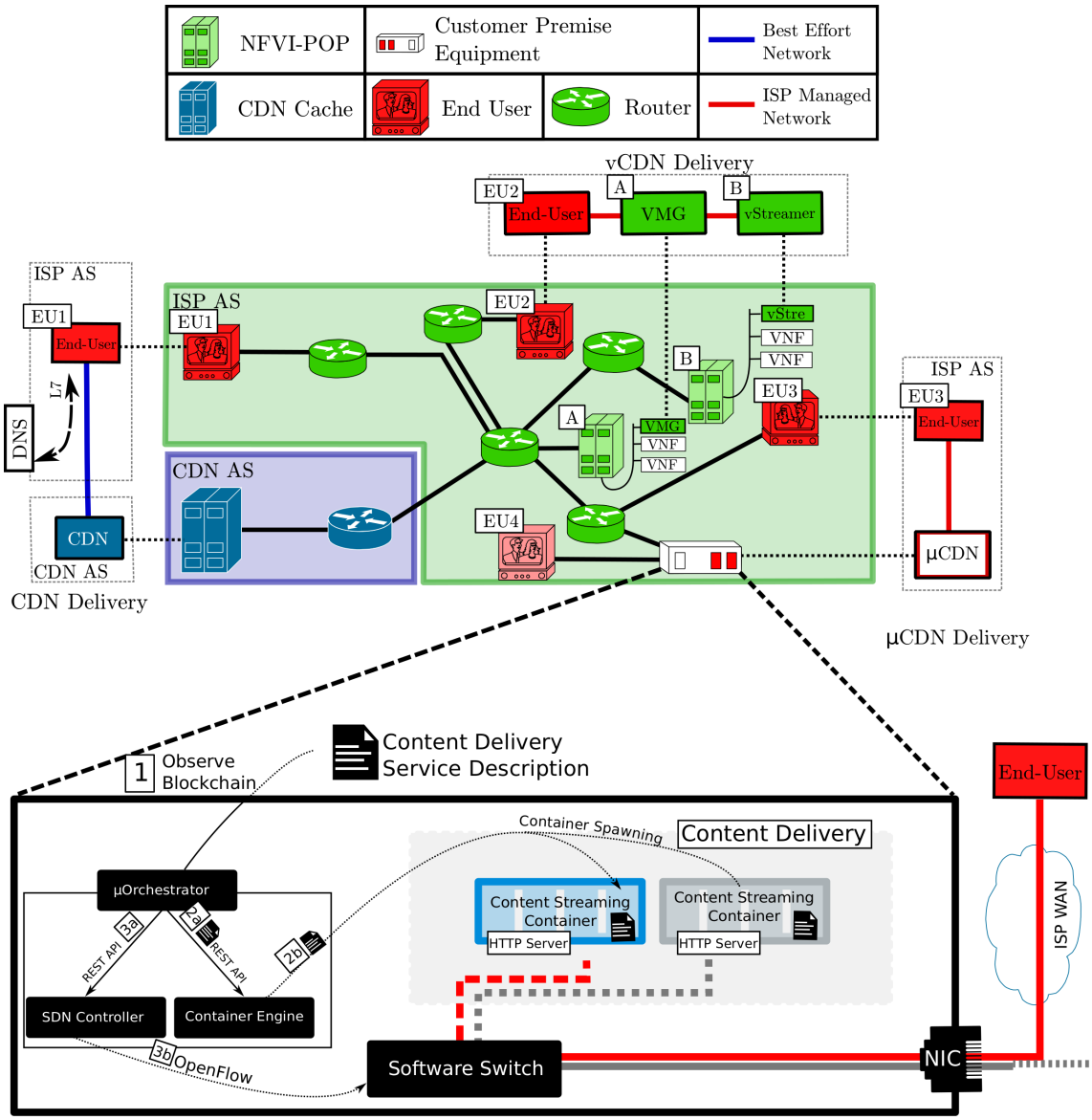
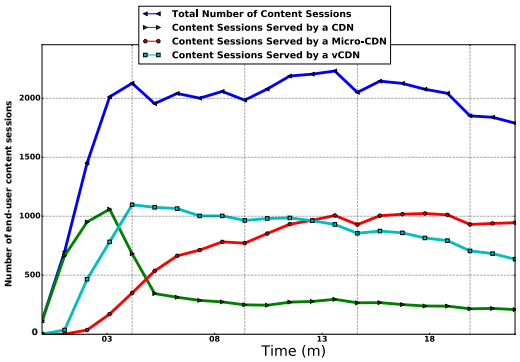
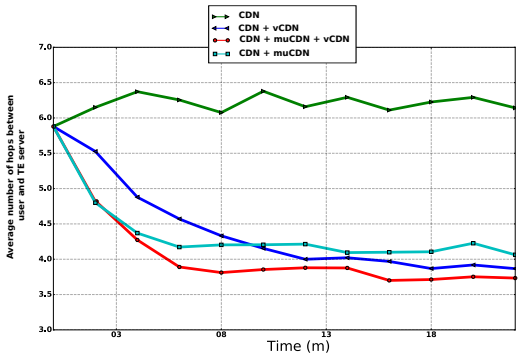


Figure 4: CDN, vCDN and μCDN services deployed in an ISP network

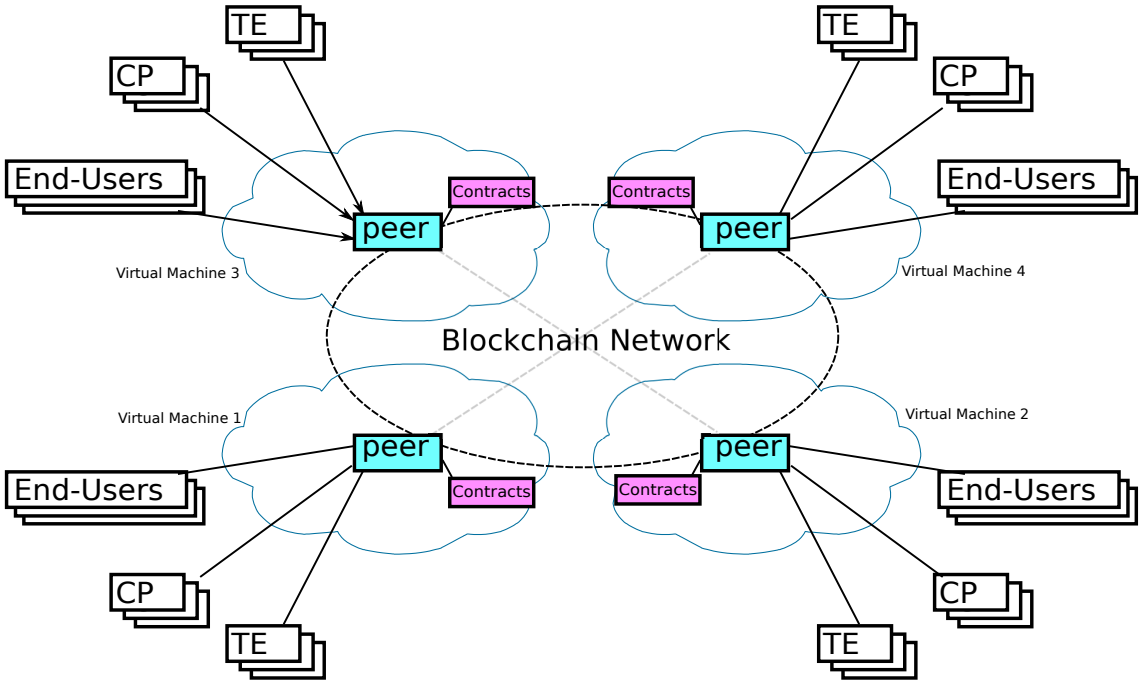


(a) Respective TEs share for CDC

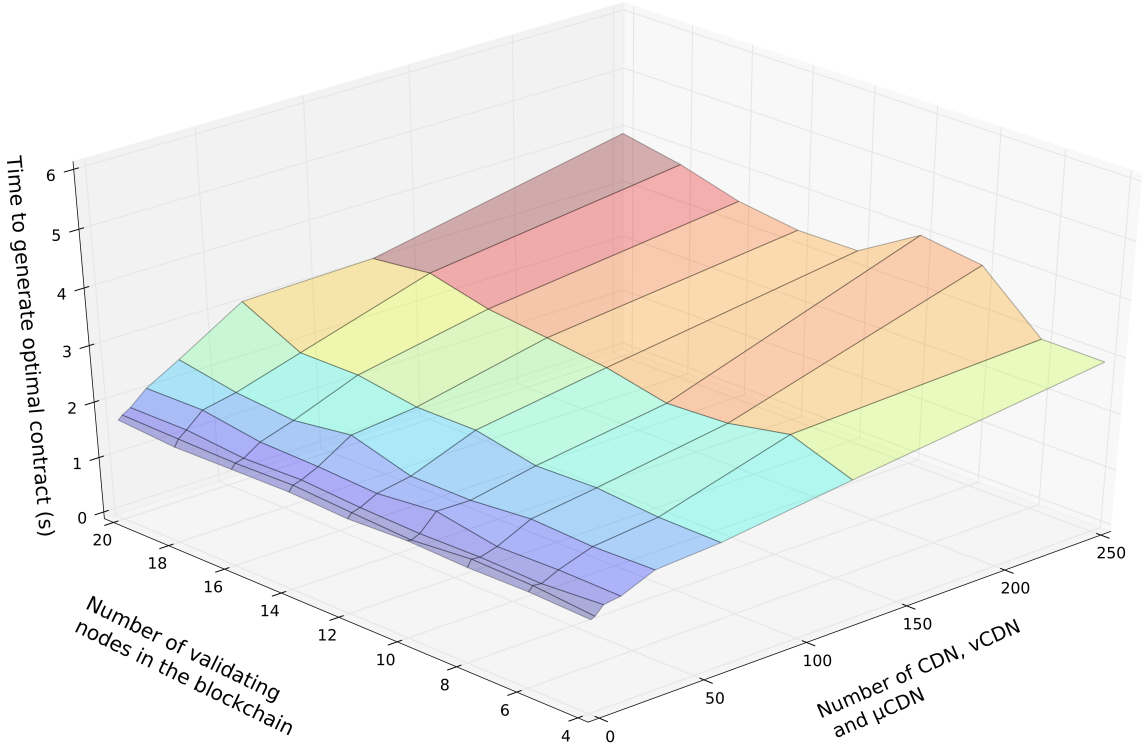


(b) Average price for content delivery

Figure 5: Network Services Chain evaluation



(a) Testbed



(b) Performance and scalability experiment

Figure 6: Blockchain evaluation