UNIONE EUROPEA
Fondo sociale europeo

REGIONE AUTONOMA DELLA SARDEGNA

FSE 2014-2020
POR
SARDEGNA

Università degli Studi di Cagliari

PHD DEGREE
Electrical and Computer Science
Cycle XXXII

TITLE OF THE PHD THESIS
Quality of Service improvements for real time multimedia applications using next generation network architectures and blockchain in Internet Service Provider cooperative scenario

Scientific Disciplinary Sector(s)
ING-INF/03

PhD Student: Michele Scarlato

Coordinator of the PhD Programme: Prof.Alessandro Giua

Supervisors: Prof.Luigi Atzori, Dr.Cristian Perra

Final exam. Academic Year 2018 – 2019 Thesis defence: January-February 2020 Session

Contents

1 Introduction

Real time communications are becoming part of our daily life, requiring constrained requisites with the purpose of being enjoyed in harmony by end users. The factors ruling these requisites are Quality of Service parameters which are strictly related to the users' Internet connections.

Achieving a satisfactory QoS level for real time communications depends on parameters that are strongly influenced by the quality of the network connections among the Internet Service Providers, which are located in the path between final users and Over The Top service providers that are supplying them with real time services.

Final users can be: business people having real time videoconferences, or adopting crytpocurrencies in their exchanges, videogamers playing online games together with others residing in other countries, migrants talking with their relatives or watching their children growing up in their home countries, people with disabilities adopting tecnologies to help them, doctors performing remote surgeries, manufacturers adopting augmented reality devices to perform dangerous tasks.

Each of them performing their daily activities are requiring specific QoS parameters to their ISPs, that nowadays seem to be unable to provide them with a satisfactory QoS level for these kinds of real time services. These kinds of communications are not only requiring a certain level of QoS, but also their privacy and security have to be guaranteed. Through the adoption of next generation networks, such as the Information Centric Networking, it would be possible to overcome the QoS problems that nowadays are experienced.

By adopting Blockchain technologies, in several use cases, it would be possible to improve those security aspects related to the non-temper ability of information and privacy. I started this thesis analyzing next generation architectures enabling real time multimedia communications [1]. In Software Defined Networking, Named Data Networking and Community Information Centric Networking, I highlighted potential approaches to solve QoS problems that are affecting real time multimedia applications.

During some experiences abroad, such as the one at Murcia University in Spain and at Gwangju Institute of Science and Technology (GIST) in South Korea, the deployments of Named Data Networking gateways in the NDN worldwide distributed testbed, and a CICN in a local testbed, have been performed. Both approaches, even though they are providing promising mechanisms such as multicast and cache, which would help in the delivery of the desired QoS to the final users, are still not fully satisfying the requirements of real time applications.

During my experiments I found that applications able to transmit high quality videos, such as 4k or 8k videos, or to directly interact with devices AR/VR enabled are missing for both ICN approaches. Then I proposed a REST interface for the enforcing of a specific QoS parameter, the round trip time (RTT) [2]. In that work, I took into consideration the specific use case of a game company that connects with the same telecommunication company of the final user.

Supposing that the proposed REST APIs have been deployed in the game company and in the ISP, when one or more users are experiencing lag, the game company will try to ask the ISP to reduce the RTT for that specific user or that group of users. This request can be done by performing a call to a method where IP address(es) and the maximum RTT desired are passed. I also proposed other methods, through which it would be possible to retrieve information about the QoS parameters, and exchange, if necessary, an exceeding parameter in change of another one.

The proposed REST APIs can also be used in more complex scenarios, where ISPs along the path are chained together, in order to improve the end to end QoS among Over The Top service provider and final users. In order to store the information exchanged by using the proposed REST APIs, I proposed to adopt a permissioned blockchain, analizying the ISPs cooperative use case with Hyperledger Fabric [3].

In this work I proposed the adoption of the Proof of Authority consensus algorithm, in order to increase the throughput in terms of transactions per second. In a specific case that I examined in this work, I am proposing a combination of Information Centric Networking and Blockchain, in an architecture where ISPs are exchanging valuable information regarding final users, in order to improve their QoS parameters.

In this work I proposed my smart contract for the gaming delay use case, that can be used to rule the communication among those ISPs that are along the path among OTT and final users. An extension of this work can be done, by defining billing costs for the QoS improvements. The main produced results can be summarized as follows: the study of next generation network architectures [3], the proposal of a REST interface to enhance end to end QoS [2], the design of a permissioned blockchain for an ISPs cooperative scenario [3] and the definition of a side chain smart contract for an ICN blockchain, that still is under reviewing process.

Michele Scarlato gratefully acknowledges Sardinian Regional Government for the financial support of her/his PhD scholarship (P.O.R. Sardegna F.S.E. - Operational Programme of the Autonomous Region of Sardinia, European Social Fund 2014-2020 - Axis III Education and training, Thematic goal 10, Investment Priority 10ii), Specific goal 10.5.

Part of the activities have been also funded by the Erasmus Mundus Action 2 Project TEAM Technologies for Information and Communication Technologies, to support the first period of mobility at Gwangju Institute of Science and Technology (GIST), the PlaceDoc scholarship to support the mobility period at Murcia University, the GlobusDoc scholarship to support the second mobility period at GIST.

The thesis is organized as follows: in section II an introduction to the QoS parameters that may influence real time communications has been carried on, in section III an overview of existing next generation network architectures and the proposal of my REST interface are presented, in section IV the design of a permissioned ISPs blockchain for the end to end QoS enhancement is shown, while in section V the design of an ICN blockchain with the definition of a smart contract is described. Finally in section VI conclusions and future works are drawn.

2 A review of existing Next Generation Architecture for Real Time Multimedia Applications

In the last two decades many real time networked applications have emerged on the Internet. In order to allow these types of applications to function properly, a special support from the underlying network such as reliability, timeliness, and guaranteed delivery, as well as different levels of service quality is required. Unfortunately, the current "best-effort" Internet architecture is not able to provide this kind of support. [4]

This is the main reason that has driven to investigate on what Next Generation Network architectures could be able to face this problem.

An analysis of next generation network architectures enabling real time multimedia communications is presented in this section. Current network architectures are not able to support real time multimedia communications for immersive application exploiting virtual or augmented reality. After the presentation of the state of the art, a discussion of open issues and future research direction is presented. Attention is devoted to the analysis of the benefits that can arise when using Information centric networks (ICN) as main component of the network architecture for assisting the deployment of AR/VR.

Designing applications in which multimedia data is delivered and rendered in real time, providing a natural and realistic sense of immersion for the user, is a very challenging problem. Each component of the communication chain such as, for example, data acquisition, data coding, data transmission, data processing, data rendering augments the system complexity, increases the point-to-point delay of communications, introduces errors.

A major problem is related to the high unpredictable behavior of current communication networks which goes against the requirements for immersive and interactive multimedia communication. Novel technologies in acquisition, coding, processing, and rendering are making high quality and realistic content available. Array video cameras, light field camera, holoscopic display, high technology head mounted display, and see through lenses are some of the technologies that could drastically change the consumer and industry market in the near future [5] [6] [7] [8] [9].

Nonetheless, such technologies generate a huge amount of data that makes even more complex the design of interactive and real-time communications on current network architectures. Next generation network architectures and information centric networking are promising infrastructures for the development of immersive and interactive applications based on augmented reality (AR) and virtual reality (VR) [10].

In fact, it has been highlighted that current AR/VR applications are requiring network resources that are difficult to meet. Therefore, the network infrastructure needs to be re-evaluated to support such applications. Some well-known issues in communications are: transmission mode (unicast, multicast or broadcast), latency (lag, jitter), bandwidth, and distributed computation.

An interesting potential solution is represented by ICN architectures, that enhance the content delivery using features that the current IP network is not offering, such as data caching and the request or the route of content at network layer using the name of the content instead of the host's address.

ICN makes it possible to operate with content at different levels. Thanks to the representation used to identify a content, that is done through an explicit beginning and end semantics, it is easily possible to provide the proper amount of resource for the flow, and track how much data has passed through a device.

Providing caching, each device in the network enables the network to be content-oriented, storing those contents, indexed by content name, which can be requested by neighbor nodes. The caching is transparent to the final users, that are agnostic of the localization of the content. This approach reduces latency in the access of the content and improves network efficiency.

Multicast is very useful in some applications of AR and VR, but it is not possible to exploit it, because ISPs are usually blocking multicast traffic that is not explicitly used by themselves internally. Instead, by using ICN, it is possible to exploit this feature, that it is very helpful in use cases such as sports and shows where many users will be participating in receiving the same information.

Moreover, such approach fits perfectly even for novel viewing experience such as multi view or view interpolation. This chapter aims at reviewing future networking technologies enabling the development of immersive and interactive communication, using high technology devices which provide high quality and realistic content.

2.1 Next Generation Architectures reviewed

C.Westphal in [10] deals with the network implications of virtual reality (VR) and augmented reality (AR), in particular considering some use cases where they can be deployed. As use cases, he considered office productivity and personal movie theater, retail, museum, real estate and education, sports, gaming, medical and therapeutic maintenance, augmented maps and directions, facial recognition and teleportation.

In his work, the use of 5G networks has been considered in order to satisfy the network requirements for AR/VR applications. In fact, with 5G it is possible to have: 300 Mbps of downlink and 50 Mbps of uplink, end-to-end latency of 10ms, mobility that can vary from 0 to 100 km/h, allowing the access to thousands of users per km^2. But considering that human can process 5.2 Gbps of data based on their physical characteristics of human perception, the bandwidth requested for applications that interact with reality is still far from being provided .

ICN has been taken into consideration by Westphal for the promised abstractions offered for video delivery. The RFC7933 [11] highlights the challenges and potential of ICN for adaptive rate streaming. Furthermore, a lot of existing works on ICN can be applied in order to satisfy the QoS requirements of the AR/VR applications, for example in [12] an architecture has been proposed, that aims to support metadata driven services in an ICN, providing a better utilization of network resources through the use of metadata driven traffic engineering.

In their work, they augmented the control plane of a Software Defined Network including a level for a content management that supports traffic engineering and firewalling, avoiding implications at the application level. The architecture that aims to support metadata driven services in an ICN that they proposed, proves to provide a better utilization of network resources. They took into consideration only the content sent over HTTP that represents much of internet traffic. Some changes to the standard OpenFlow protocol must be made in order to permit the implementation of the architecture proposed.

Related to multicast, it is important to mention Network Coding (NC) [13] as technique that is very effective for collaborative media streaming applications. Substantially, the main logic that resides behind NC is the following: thinking about a system that performs as relay of information, that can be a node in a peer to peer network, a router, or a node in an ad-hoc network, when it is acting as relay it is just forwarding a packet to some other nodes that need to receive it.

Using network coding, the node is able to concatenate packets that it has received or created, generating one or more packets destined to another node.Another possibility is to combine packets linearly, that is a technique that can be used as form of information spreading.In fact, combining packets of length L linearly, that resulting encoded packet, also, will have size L.

Unlike concatenation, from the generated packet it is not possible to date the original information, but, instead, it is possible to use it in order to carry information about several original packets. In [14] Fiandrotti et al. address the problem of finding the packet scheduling policy that maximizes the number of media segments recovered in the network.

Once they cast this into a distributed minimization problem, they proposed a heuristic solution that makes the framework robust to infrequent or inaccurate feedback information. In the experimentation that they conducted in their local testbed and in PlanetLab, it has been shown that their scheduling framework achieves lower bandwidth consumption, lower playback delay and better media quality than a random-push scheme.

To reduce the bandwidth requested for the transmission of 3D immersive video, Cesar et al. [15], presented a generic real-time time-varying point cloud codec. The codec is suitable for mixed reality applications in which 3D point clouds [5] are acquired at a fast rate. In the codec, intra frames are coded progressively in an octree subdivision. In order to exploit further inter-frame dependencies, they present an inter-prediction algorithm that partitions the octree voxel space in N x N x N macroblocks (N = 8, 16, 32).

In order to transmit a frame with the octree-based point cloud compression, the requested bandwidth can vary from 40 to 265 KB, against the bandwidth required using the RGB-D coding that can range from 132 to 800 KB, achieving an optimization in the compression that can be considered around to 4:1.

Al-Shuwaili et al. in [16], proposed an approach to gain mobile energy consumption, compared to conventional independent offloading across users, implemented via Successive Convex Approximation (SCA). As use case, they took the class of AR applications in which artificial images are overlapped to the real world, via the screen of a mobile device.

The approach is expected to be composed of five elements: Video source, Tracker, Mapper, Object recognizer and a Renderer. The only components that need to be executed in the mobile devices are the video source and the renderer, that respectively serve to receive the raw video frames from the camera of the smart phone, and to show the frames that have been processed through the display.

Meanwhile, most of the computation will be performed in the Tracker, Mapper and Object recognizer, that perform the functions of tracking the position of the user related to the environment, building a model of the environment and identifying objects that are already known. These three components can be offloaded, and if it is, Mapper and Object recognizer can collect inputs from other users that are in the same area. In this work, it is shown the collaborative nature of the AR applications, necessary in order to reduce communication and computational overhead.

Ravindran et al. [17] proposed a framework to enable ICN based service platform as Virtualized Network Functions (VNFs) and also to enable several edge-cloud services such as enterprise applications, big data analytic, or M2M/IoT services. The platform is generic, and is able to support several ICN protocols and corresponding real-time and non-real-time services, leveraging ICN features such as name based routing, caching, multicasting, and flexible security techniques. They compare the scalability among the proposed ICN platform and a peer-to-peer design, making a performance analysis on a network based conferencing solution for an enterprise that is aimed for interactive real-time multimedia applications capable of scaling to many participants.

Ravindran et al., in [11] are proposing a framework called HomeCloud, with the purpose to be more efficient in terms of latency, able to support the massive and ever-increasing amount of data generated by the 'Internet of Things' and improving the portability across platforms, through the use of open standards, such as NFV and SDN, and the consequential implementation of VNFs. To break the monopoly of the conventional centralized cloud computing providers, and to nurture innovations, especially from the small and medium-sized application providers, they are proposing an open edge cloud framework to efficiently deliver future new portable applications over a shared edge infrastructure. The framework uses NFV in the edge, affording local computing, storage and networking close to edge mobile devices, that are running those applications that generate massive volume of data and require low latency.

Humernbrum et al., in [18], proposed an architecture design of an SDN module which implements the API functionality required by Real-time Online Interactive Applications (ROIA). Even if the SDN module has not been proposed to support AR or VR scenarios, it is suitable for computation and interaction-intensive training, multiplayer online games, simulation based e-learning, and more. All these applications demand high quality of service (QoS) on the underlying network. In order to meet these requirements, they proposed to use SDN, for exploiting the decoupling of control and forwarding logic, from the network infrastructure, making it programmable for applications.

The aim of the module is to help in the management of the SDN infrastructure at runtime according to the application requirements, in order to lead to a higher and better quality of experience (QoE) for the end-user. QoE has been defined as a suboptimal QoS perceived by the end-user.

In-home consumer electronic (CE) devices that incorporate the emerging SDN

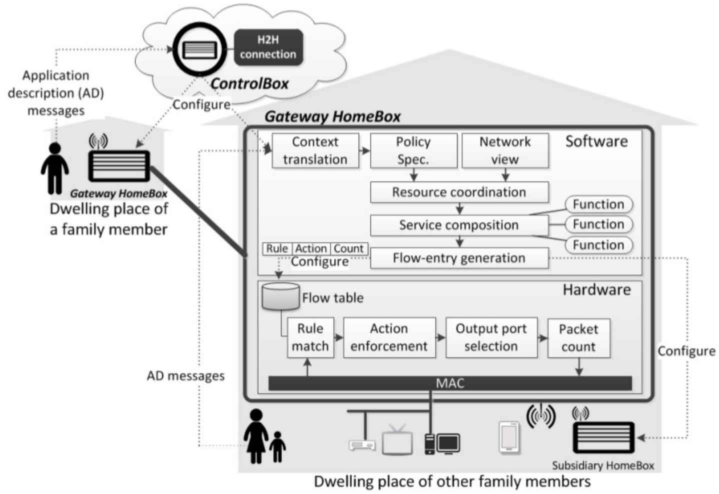

Figure 1: System architecture of Gateway HomeBox and the SDN-leveraged interconnection of two home networks [19].

paradigm have been shown by Kim et al. [19]. Through the prototype that they developed, they realized a multi-home visual-sharing scenario, tying family members closely, by geographically extending their home networks. In this scenario, for example, children can remotely enjoy a film which is provided by a home media center at their relatives. The network architecture that they design, is composed of two main components: a Control Box and an SDN-enabled Homebox Gateway.

In order to interface users with the devices, an Application Description set of messages has been defined. The control box has the task to configure the HomeBox, in order to support the integration of multiple home networks. The federation task is performed only when a user sends explicitly a request with an AD messaging. It is important to say that the Control Box will be hosted, with high probability, by third party trust provider. For this reason, an SDN-based two-tier messaging has been implemented, in order to prevent privacy concerns about opening the packet visibility, to a central point.

The raw information of every IP packet within a home network is exclusively open to the HomeBox devices deployed in the same network; thus, the ControlBox cannot directly take in-home packet visibility apart from getting fine-grained networking information from a user-generated AD message.The HomeBox Gateway can be thought of as a small-scale SDN controller having a rich featured data plane, which accomplishes a flow-based virtual overlay networking.

Within a home it functions as a residential home router but provides a mission-critical networking service, alone or in cooperation with a subsidiary HomeBox. This device has the packet visibility of all in-home flows and leverages a home network view to compose a network service. The service composition can be performed through a pipeline in which modularized software functions are routed.

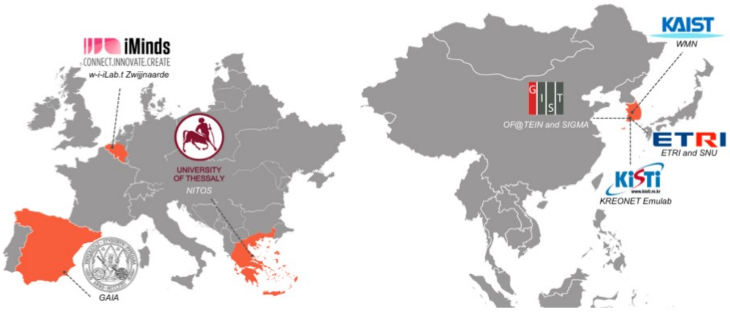

Figure 2: The SmartFIRE federation of European and South Korean testbeds [20].

The HomeBox is also responsible for populating flows entries to be fed into the embedded data plane or sent to the subsidiary HomeBoxes. The flow entries are stored in the flow table, that guides 'what and how' to process packet flows. Even if it is considered a home network scenario, the use of a flow-based networking control on a vantage point enables improved control granularity and major flexibility. Figure 1 shows the system architecture of the Gateway homebox, and shows how through the AD messages the home to home connection is performed via the ControlBox. It is important to mention the possibility to implement in an intercontinental SDN-based testbed the architecture that I am going to propose in this work.

SmartFIRE [20] includes many smaller-scale testbeds in Europe and South Korea. It is the first intercontinental federation of SDN, that provides wireless, wired and cloud technologies. The aim of the framework is to provide experimentation services easily with the heterogeneous resources that have been made available. The infrastructure is OpenFlow-based, and experimenters can access the infrastructure just once they have been authorized and authenticated through a distributed Public Key Infrastructure (PKI).

PKI permits users to access to the different federated testbeds through a single account, while resources' reservation has been organized using Slice-based Facility Architecture (SFA) [20] that is a GENI adopted architecture. The main purpose of SmartFIRE is to extend the cOntrol Management Framework (OMF) [21] unifying multiple testbeds in just one platform and controlling them from a single framework.

It is in fact taken into consideration the possibility to test an ICN architecture, that has been already implemented in SmartFIRE, using wireless and SDN technologies. Figure 2 shows which testbeds form the federation of the European and South Korean SmartFIRE.

Table 1 shows a list of the problems, the network technologies and the use cases addressed in the related work analysis.

Table 1: List of Problems, Network Technologies and Use Cases Addressed in the Related Works.

Ref	Problem addressed	Network Technologies	Use case
[10]	Challenges in Networking to support AR/VR Applications	5G ICN MEC SDN	General
[17]	Provide edge-cloud services for M2M/IoT, enterprise application and big data analytic	Edge-Cloud ICN NFV SDN	Conference
[22]	Provide robustness and efficiency in data dissemination using distributed data synchronization	NDN	Chat
[18]	Provide high QoS in SDN, using an SDN module which implements API functionality requested by ROIA	APIs SDN	ROIA
[19]	Realize a multi-home visual-sharing scenario extending geographically home networks.	SDN HomeBox and ControlBox (prototyped devices SDN-Enabled)	Multi-home Visual-sharing
[16]	Reduce mobile energy consumption under latency constraints, through a joint optimization of the allocation of communication and computational resources.	MEC	AR
[23]	Provide a federated intercontinental testbed with wireless, wired and cloud capabilities.	ICN MEC NFV SDN	Defined by the experimenters
[14]	Improve the throughput and provide a high degree of robustness through the linear combinations of previously received information by one node of the network.	Network Coding	General

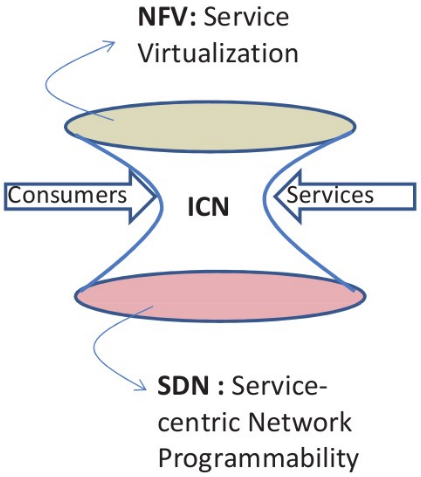
NFV: Service Virtualization
Consumers
ICN
Services
SDN : Service-centric Network Programmability

Figure 3: ICN as A Service-Centric Narrow Waist [17]

2.1.1 An ICN architecture based edge-cloud services

In [17], Ravindran et al. wanted to exploit the features offered by Network Function Virtualization (NFV) and SDN. NFV offers Open-APIs that permit the communication among the cloud computing framework and the outside world. Through these APIs it is possible to achieve service virtualization allowing dynamic scaling of computing. Figure 3 shows a graphical representation of the relation between NFV, ICN, and SDN using the narrow waist model.

NFV can be run on commodity hardware, ideally providing functions that work from L2 to L7 of the ISO-OSI stack. Challenges in NFV range from realizing software functions as reliable, secure, and with the same performance as hardware based realizations, to complex live migration and scale network functions on-demand across multiple domains. They consider that software functions not only need to be integrated with existing network management infrastructure, but also new management extensions are required, in order to monitor the virtualized infrastructure.

On the other hand SDN offers the separation of control and forwarding planes, permitting fine grained service control, adaptation to varying states of the forwarding plane or states of a service. This is done through the use of standardized APIs that connect services and controllers, named Northbound APIs, and controller and forwarding elements, named southbound APIs, that allow independent evolution of the service, control and forwarding planes.

Challenges in SDN are mainly related to the controller management, in particular it is necessary to address topic, regarding the northbound boundary such as: placement, fault tolerance, consistency of networks state and forwarding policies; while for the southbound interface the challenges are about the scalability of the forwarding elements, to handle flows from various work loads arising from deployment scenarios, such as in an enterprise, data center, or provider environments.

The architecture of the proposed framework has been represented as an hourglass. On top of it, there is the service virtualization layer enabled by NFV, that permits the provisioning, scaling and migrating on demand services, that in this case are represented by ICN protocol and service instances. At the bottom of the hourglass there is the SDN, used to manage the bandwidth, permitting, in fact, to subject ICN flows. Those flows are related to consumers, ICN services communications and related to service policies, which also include context adaptation.

The waist is the ICN, that connects consumers to services through expressive ICN-APIs that are extended to express contextual changes, in order to allow real-time adaption of services to the consumer's context(s).

They proposed an hourglass representation of the ICN as A Service-Centric Narrow Waist architecture, where at the bottom of the hourglass there is SDN which allows ICN flows between consumers and ICN services to be subjected to service policies, which also include policy adaptation. SDN will be used to satisfy the dynamic bandwidth requested from ICN services, that certainly leverages computing and storage resources.

ICN is, in fact, in the waist of the hourglass, in which consumers and services are connected through expressive ICN-APIs that not only include Get() or Put() primitives, but also extended primitives to express contextual changes, in order to allow real-time adaption of services to consumer's context(s). At the top of the hourglass there is the service virtualization layer enabled by NFV, which allows services to be provisioned, scaled and migrated on-demand. Referring to ICN, the services are ICN protocol and services instances.

They proposed an ICN based edge-cloud framework, in order to exploit some of the features that ICN offers, such as: the in-network processing, large scale mobility through late-binding features and equal support for both, ad hoc and infrastructure based application and the transport layers. ICN de-couples application from the transport layer, by first naming entities such as applications, services, and content and then binding consumers to them, through ICN's name resolution layer.

The framework that they proposed is aimed for:

- Information-centric application such as conferencing or IoT applications, and provides name based content dissemination, in-network caching, receiver-oriented interest and data multicasting, content level integrity, privacy and provenance.

- The provision of open-APIs to stake holder entities, such as ICN service owners (i.e. if service is hosted), ICN service controllers (for policy fine grained enforcement, performed from operator's or third party controlled) and ICN consumers (consumers of a particular ICN service, such as IoT applications, conferencing, V2V etc.).

- The adaptation to user dynamism, through the adaptation to the change of the context and to achieve service level dynamism, such as scaling, migrating, and replicating service resources on-demand.

The important components that form the architecture are: ICN cloud orchestrator, ICN service platform and ICN service layer. The ICN Cloud Orchestrator is the interface with the ICN service owners, that are able to program the cloud resources through the ICN service-API. These APIs allow service owners to express ICN specific change to computing/storage/bandwidth resources to adapt to service load conditions.

The orchestrator converts service requirements into actions that permit to meet service objectives: this is done through the communication with the ICN service controllers that, in turn, interact with the underlying NFV cloud, composed of ICN Service Platform. The ICN service controllers are one of the components that realize the control plane of the SDN framework, together with the ICN network controller.

These controllers interact via the ICN service control API. Through the service-API, the platform provides feedback of service usage statistics to service owners. This permits to scale service requirements benefiting through a pay-as-you-go strategy. Furthermore, service customers (as in an enterprise) use the service-API to provide the required resources as in a software-as-a-service (SAAS) set up, in which case a slice of the ICN service resource is devoted to the customer's use.

The ICN service platform is composed of VNF instances in the form of ICN protocol and service control functions. The protocol is implemented through an ICN service router, that activates an ICN forwarding plane; this, in turn, performs name based routing inter-connecting ICN services instances, which can be distributed among several edge-cloud instances. The forwarding plane is managed through the ICN service controllers aforementioned, that manage the name based routing policies of the ICN service router.

The control plane is composed also of ICN service gateway (ICN-SGW), which is the interface with ICN consumers through the ICN UNI-API, to resolve service requests to ICN service instance(s). The UNI-API are used to help in service discovery and service context adaptation of the User Entity (UE). Another component of the platform is the Service Profile Manager (SPM), that is a database used to resolve ICN service ID to the location(s) of the ICN service instances.

ICN services are instances of applications such as content distribution, conferencing, or IoT services, which execute their own service logic through interaction with several other service instances. The ICN service layer implements service functions in order to allow the consumer's application to interact with the ICN service platform, using the ICN UNI-API.

The End User device and the home router of the user, interact with an architecture in which a Service Access Layer is implemented, with the aim of providing functions such as: service discovery, resolution, publish and context management. In the same architecture, applications interact with SAL through the ICN APP-SAL API, at least in the phase of service bootstrapping and management functions.

Inside the ICN service gateway there is a service access point (SAP) that interprets user requests and invokes service management tasks, which include discovery, resolution, publishing, or handling context changes. These tasks can be handled locally or with the help of other SDN control components and the SPM. Instances of ICN service can trigger service orchestration through ICN-SDW's service management functions.

Service orchestration is the concatentation of multiple services realized as a service graph to satisfy a user request. The interaction among service management functions of ICN-SGW could also be sued by ICN services in order to discover other services, or to request more resources on-demand, so as to meet its service requirements without involving its own service control components or the service manager.

The ICN service platform that they proposed exploits cloud computing features of NFV in order to realize an ICN platform that is able to instantiate several services corresponding to any ICN protocol. By nature, the instantiated service could be real-time or non real-time. A network based conference application has been considered as reference use case.

They measured the convergence time and compared, through a performance analysis, with a peer-to-peer design model called Chronos [22]. The network topology has been simulated using ndnSIM [24], and it has been designed including two segments of the transport network: an access network and a core network. For the core network two different topologies have been considered, a 3-by-3 grid network, and an Abilene network, in three different simulation scenarios.

The access network is composed of a two-level tree topology, in which each access network has a router connected directly to a proxy, that resides in the core network. The use of proxies introduces less control overhead in the network based solution compared to the peer-to-peer model. In fact, the update load handled by a proxy node increases linearly with the number of proxies. This is in contrast to the peer-to-peer case where the total of control messages in the network is $O(n^2)$, where n is the number of participants in the network.

2.1.2 ROIA architecture

Using SDN it is possible to avoid static techniques of controlling the QoS, like the reservation of network bandwidth with Resource Reservation Protocol (RSVP) or Diffserv, that have to be configured manually by the network administrator, as they do not fit with the dynamically changing demands of ROIA.

The specifications, the design and the implementation of a novel Northbound API for the development of ROIA, which can use the advantages of SDN, are also described. Experimental testing and the evaluation of the results, for its prototype implementation are finally reported. Figure [18] shows the basic architecture, in which it is possible to observe how ROIA client are directly connected to a SLA Manager, in order to be provided with the requested QoS, and the SLA Manager communicates with the SDN Module running on the Server through the SDN Controller.

The SDN module is composed of three components: a) the specification component which offers data structures and functions which are used by the ROIA developer to formulate network requirements. In the SDN module, a flow is uniquely defined by the sender's and receiver's IP address and port, as well as an optional flow label.

b) The administration component that contains the key functions of the SDN module. Through these functions the ROIA developer can transmit QoS policies to the SDN controller or cancel the requirements of policies that have been already transmitted to the controller. Implementing this module outside the controller, permits the ROIA developer to inquire SDN module in order to know the current status of the network without having direct contact with the controller.

c) The communication component is used to coordinate the connection and communication among the application and the SDN controller. This component

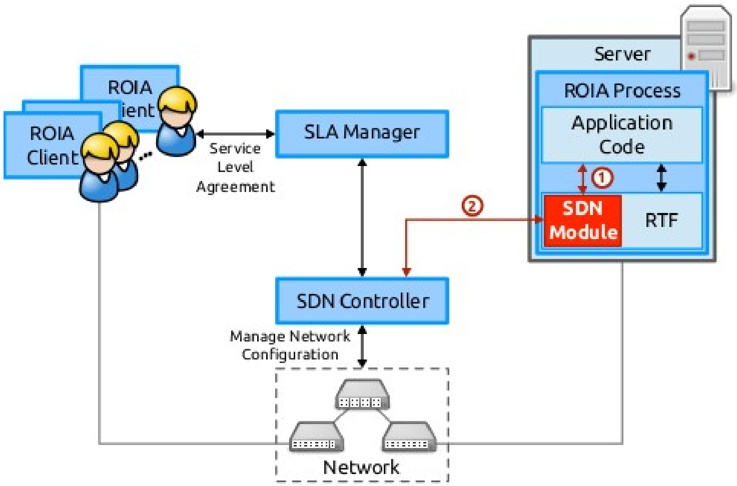

Figure 4: Architecture: a ROIA process serves connected clients, re-quests network QoS from the SDN Controller, and reports QoS information to the SDN Controller [18].

works transparently for the user, in background, neither ROIA developer works with this component. The communication among the specific implementation language of ROIA API and the controller side is performed through a REST-based API implementation.

Taking an upper level look at the architecture, it is possible to identify the four main components, that are: a ROIA process, ROIA clients, the SDN controller and the underlying network. In the ROIA process the SDN module is implemented and it will interact with the SDN controller. In the tests performed, in order to provide a good QoE to the final users, they considered the requested bandwidth as the only metric.

In particular, it has been showed that the controller is able to react to the situation in which the hosts start to increase the traffic, and are using the path already used by the application, that requires a minimum amount of bandwidth in order to provide a real time service. The simulations have been done using Mininet [25], and in particular, the increase of traffic is performed through the use of Iperf [26], that is able to continuously send randomly created data to a given receiver and calculates the achieved throughput.

2.2 Discussion of future research direction

Westphal in [10] specifies some requirements on the network architecture. Usually network architectures are hierarchical server architectures, where functions are provided to the edge of the network, in order to ensure responsiveness and accomodate the users in accordance with their number. Another element is needed, which is the

Virtual Environment (VE). This element can be shared among clients, in peer-to-peer applications.

The introduction of this element brings to a new architectural design whose goal is to place the processing as close to the edge of the network as possible. Westphal considers the 5G scenario, and identifies in the Mobile Edge Computing (MEC) [16] one possible network architecture, where MEC server holds AR data and AR object caches for high bandwidth/low latency content delivery.

Furthermore, it is considered the use of APIs and properly placed functions in order to transmit the representation of a VE. One solution can be the transmission of the whole representation, and let the device choose which specific view to display. But this solution is the most inefficient way to use the network, because the representations that will be transmitted, will be never observed by the user.

In [27] it is treated the specific sub-case of 360 video in which it is demonstrated that the approach that they propose can save up to 80% of bandwidth consumption. They proposed two methods: the first is packetizing the representation of the VE into views, and then sending only the views required; while the second method is the prediction of what direction the user will take, forecasting the position, and finally using this prediction in order to send just the predicted views.

Ravindran et al. in [17], proposed an ICN framework implementing a chat application in which they show how more scalable their hierarchical proxy architecture is compared to Chronos [22], that uses a peer-to-peer approach over NDN [24]. The original challenge was a conference solution, in fact they proposed to prototype their architecture for enterprises that want to exploit interactive real-time multimedia applications. They use Content Centric Networking (CCN) whose goals are to provide more secure, scalable and flexible networks; an interesting derivative of CCN is Named Data Networking (NDN), that, thanks to the architecture differences integrated, reduces the time in search of data.

Humernbrum et al. in [18], in order to address the dynamic QoS demands of ROIA, proposed a novel Northbound API, that has been specified, designed and implemented. The functionality of the proposed APIs is implemented in an SDN module, and it has been tested with mininet [25], that is a network simulator. In order to be more precise and realistic in measurements, the SDN module can be tested in real testbeds, such as, for example, the above mentioned SmartFIRE.

Kim et al. in [19], prototyped SDN-enabled CE devices that permit to create a multi-home visual-sharing network. A combination of their prototype together with an ICN architecture in order to provide caching of the contents, can be an interesting adaptation so as to allow the deployment of multi-home augmented reality applications. This integration would require more disk space than the one offered by the FPGA used in their prototype, hence an upgrade to a bigger disk would be mandatory.

Coding techniques have been taken into consideration in this work, such as the codec presented in [15] and the possibility to implement Network Coding in the data plane of the architecture that I want to implement. It is in fact clear, that the solutions nowadays proposed still are not satisfying the strong requirements of real time multimedia applications.

2.3 QoS Architecture

In "Leveraging OTT and ISP cooperation to enhance end to end QoS by exchanging valuable resources" [2] I proposed a REST interface for the enforcing of a specific QoS parameter, the RTT. Over The Top (OTT) providers detecting bad QoS for their users can not influence on the network by themselves. On the other hand, the Internet Service Providers (ISPs) along the path only see the effect in their own network. The users might agree on reducing a property (e.g., the bandwidth) on contract Service Level Agreements (SLA) with ISP to enhance other properties (e.g., Round Trip Time).

This work proposes a communication mechanism between OTT and its own ISP who could act as a relay with bordering ISPs chaining up to user premises, thus finally obtaining better QoS for the pair OTT/user, reducing ISPs load in return. This work proposes a REST interface for the specific case in which enforced RTT is to be exchanged by reducing available bandwidth as a counterpart, to be extended in future work.

Two entities exchanging information remotely is defined as end-to-end connectivity. This does not apply to worldwide communications, where connectivity is delivered over a bunch of ISPs. In general and for short term communications, the entities have only control over the last mile connectivity parameters, by means of contracts with strict SLAs with their ISPs.

If specific QoS is required for a certain service, only in special cases and mainly through third parties, long term SLAs can be signed, but for individuals and for short periods of times, this kind of service is not affordable. As a result, the deployment of services in the last mile, such as CDNs, has been lately widely employed. But for interactive services in which content is dynamic and/or calculated, the prior approach is not always feasible.

OTTs operate on top of the Internet and those relying on end-to-end communications between their customers and centralised servers, suffer from the best effort approach of the Internet. Even if the OTT might be able to identify low QoS for a certain customer which probably is not a customer from its own ISP, the OTT can negotiate better SLAs with its own ISP and obtain better communications for the whole connection, but that does not ensure that this precise customer obtains an increase in the QoS.

The OTT requires the means to request for enhancements for a certain customer which, in turn, might require extending the request to multiple ISPs in the path in a delegated QoS request. In this work I define a Representational State Transfer (REST) application program interface (API). The APIs I defined use the HTTP requests POST and GET, in order to create and retrieve resources. RESTful API is an architectural style and approach to communications that is generally used in the development of web services. The stateless nature of the calls enable this approach to scale and accommodate load changes.

On the other hand, the customers might agree on reducing some terms of their SLAs to augment others, e.g, reduce the download speed in order to reduce the RTT to a certain service, similarly to what some DSL operators offered with Annex M, by which upload speed was incremented by means of sacrificing the download speed on user request. How the changes in the network might be enforced is out of the scope of this work, but leveraging on SDN looks like a good starting point for such a system.

These dynamic agreements attract new business models in which OTTs would be able to enhance their Quality of Experience (QoE) by means of customer oriented QoS enhancement, users could experience an increase in the QoS for their favourite OTT by reducing unused parameters of their contracts and finally the ISPs, not only the one related to customer and OTT but also the intermediate ones, which could sell better paths to the edge ISPs. These last ones would be able to reduce the over-provisioning of their networks by means of these agreements with their customers, exchanging SLA capabilities in a profitable way for the ISP and with extra incoming in the case of the OTT, by offering a finer grained service enhancement.

2.3.1 Related works to the QoS Architecture proposed

Many solutions have been proposed in order to solve QoE and QoS related problems. Among them, there are solutions that make use of the Software Defined Networks (SDNs), which is a network paradigm where the programmability of the control plane can be exploited in order to monitor QoE parameters. Other solutions taken into consideration in my work have an economical approach.

The generic QoE management framework presented by Seppanen et al. [28], which is applicable to a broad range of systems, can be useful for data acquisition and the monitoring levels. In their work it is demonstrated an instantiation of the framework as a network access point management system for RTP-based video. The system is able to positively affect the perceived quality of the multimedia application considered, and also to reduce over-prioritization and optimize resource usage.

The QoE function, proposed as a "QoE-service" for on-demand services, or premium users, based on SDN, introduced by Liotou et al. [29], is able to provide a global resource view. Furthermore, the view is combined with complementary QoE metrics, to assure the desired performance for OTT applications by adopting traffic management mechanisms. Also they investigated a set of use cases that demonstrate its suitability and applicability to Long Term Evolution (LTE) networks.

An interesting management and orchestration architecture able to differentiate network services with quality level assurance and to enforce agreed SLA, has been defined by Ongaro et al. [30]. The defined architecture exploits the use of SDN in conjunction with the OpenFlow protocol. They formulate the problem of enhancing Qos and QoE in terms of packet loss and delay, with the Integer Linear Programming (ILP). In the formulation the network constraints and the requirements of real-time applications have been taken into account. Also, once discovered the optimal solution, they evaluate the impact and benefits of the proposed schema using the Mininet network emulator.

Durner et al. [31], provide a study of the impact on dynamic QoS mechanisms and their realizations for OpenFlow-enabled SDN switches. Although SDN and, in particular, OpenFlow as one dominant realization claim to provide a standardized interface to control network traffic, their measurement results show a noticeable diversity for different OpenFlow switches.

2.3.2 QoS Architecture proposal

In my proposal it is considered the general use case where both user and the OTTs perceive a lack of quality in the service. But the OTTs might not be able to influence, neither know, anything about the user's ISP connection, even if the ISP is the same for both entities.

For the OTTs, receiving information related to the connection of the users, would be very useful to figure out what the problem is. I propose a way to trigger the ISP to make an exchange of one parameter for another, for example enabling the exchange of bandwidth in excess in favor of a reduction of the RTT, in order to provide a better QoS to final users. In particular, gaming, video streaming and video-conferences are applications that need to reduce lag. I am proposing REST APIs useful for communication among OTTs and ISPs, able to show to the OTTs some QoS parameters, and verify if they are below the desired levels, while other parameters are being wasted or unused.

There is no restriction or suggestion on how the ISPs could enforce a certain QoS level and restrain the exceeding parameters in exchange, but it is clear that the SDN paradigm can play a key role in this kind of solutions, both because of the possibilities that it offers in terms of applying flow based policies, as well as the centralized control view of the network, therefore being able to assess its real status.

Table 2 describes the calls to the methods that I consider useful, in order to provide a better QoS to final users experiencing lag in the gaming use case. The simplest case I considered is when OTT and user are using the same ISP. Whenever there are one or more ISPs in the middle, the chaining has to be considered. The chaining here refers to the advantage of the connection between border ISPs which implies a trust relation so that the OTT's ISP relays the request, if accepted, to the next ISP; likewise packets are forwarded based on routing algorithms.

It is not foreseeable a direct connection between the OTT and any ISP which is not providing a direct service to the former. For example, if the game company is Spanish and connects through Telefonica, and the user is connected through the same telecommunication company, the ISP will be the same. Supposing that the ISP and the game company deployed the REST APIs that I am proposing, and the latter notices that one or more of his users are experiencing lag, the OTT will try to ask the ISP to reduce the RTT for a certain user, or a group of users.

The gaming OTT can perform a call to the method:

POST */qos/rttreduction?network=IPv4/netmask&maxrtt=maxdesiredrtt*

asking the ISP to reduce the RTT, passing as parameters the IP of the host or the network of hosts, and the maximum RTT desired.

The information about the QoS parameters of the user connection can be retrieved by the OTT calling the method:

GET */qos/minmaxrtt?network=IPv4/netmask*. This method is used to request to the ISPs the minimum and the maximum RTT for a particular host, or network of hosts. In order to have a fair exchange, reducing the lag in exchange of, for example, a reduction of the bandwidth, the OTT needs to know which is the amount of bandwidth available for one particular host, or a network of hosts. The game company, in this case, can call the method GET */qos/availablebw?network=IPv4/netmask*. If there

Table 2: REST APIs for ISP communication in QoS.

REST Call	Description
POST $/qos/rttreduction?network = IPv4/netmask\&maxrtt = maxdesiredrtt$	With this POST the gaming OTT asks the ISP to reduce the Round Trip Time, passing the IP of the host or the network of hosts, and passing the maximum RTT desired.
GET $/qos/minmaxrtt?network =$ IPv4/netmask	With this GET it is requested to the ISP the minimum and the maximum RTT for one particular host, or network of hosts.
GET $/qos/availablebw?network =$ IPv4/netmask	With this GET it is requested to the ISP the bandwidth available for one particular host, or network of hosts.
POST $/qos/exch - av - bw4rtt - red$	With this POST the OTT wants to exchange the available bandwidth with a reduction of the RTT.

is remaining bandwidth for a certain user, or a group of users, and it is possible to reduce its or their RTT, the OTT can propose to the ISP to exchange the available bandwidth with a reduction of the RTT, calling the method POST *$/qos/exch - av - bw4rtt - red$*.

In scenarios where the OTT and the user are connected to different ISPs, the chaining concept, as shown in figure 5, must be applied. It is shown how the OTT is connected to its ISP, in turn connected to bordering ISPs. The former, before reaching the users ISP, has to cross one ore more border ISPs in the path to offer End-to-End QoS. In this scenario I assume that data concerning the QoS perceived by the gaming OTT for a certain user, is measured in the OTT server and is provided to a QoS Monitoring service. This latter triggers, if necessary, a QoS Reduction via a QoS Reaction module, which in turn contacts the ISPs External Services API (ESA) that could be implemented as a standalone service whose proxy is the network controlling system (let it be SDN or any other network management system), or integrated, as part of the northbound interface of the controlling system itself. The ISPs network management system will check if it is possible to enhance the QoS in its domain and will forward a request to the next (if any) ISP in the path to the user.

A certain degree of collaboration between bordering ISPs can be assumed. Once

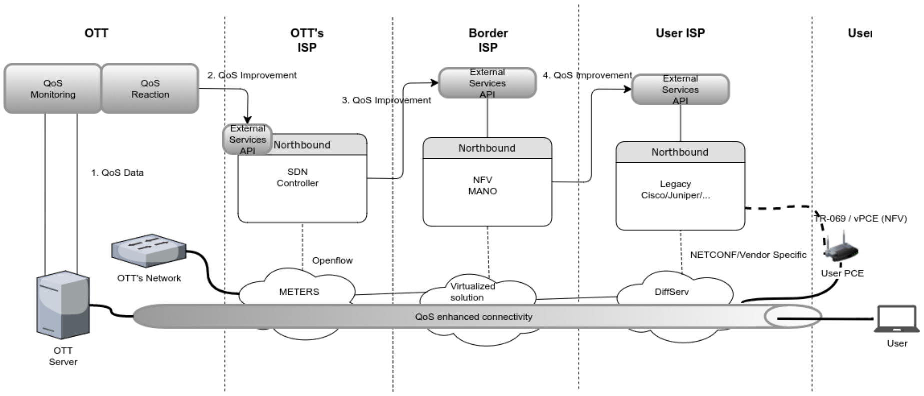

Figure 5: Chaining requests.

the users ISP receives the request and if SLAs with the user, in addition to network architecture, allow it, the QoS enhancement could be exchanged for exceeding bandwidth, be billed, or any other prestablished agreement. The figure exemplifies some of the current and probable future network architectures that are foreseeable to be employed by ISPs but does not try to be complete in that sense. So, the gaming OTT will trigger a call to the POST where each ISP in the path is asked to reduce the RTT. Also the following steps have to be performed for each ISP: the GET requesting the minimum and the maximum RTT, the GET requesting the bandwidth available, and the POST where it is expressed the will to exchange the available bandwidth, whereas there is some, for a reduction of the RTT, whenever it is possible.

This work proposes the basis and a first approach to a chained collaboration between OTTs and ISPs, focused on the enhancement of QoS by exchanging and balancing connection parameter values, so that the OTTs provide their customers with enhanced service by a reduced price or even free, the ISPs supply better QoE to the customer while reducing parts of their SLAs as a counterpart and the customer finally solving a problem with the service, being consumed almost transparently.

The RESTful methods proposed are envisioned for the use case in which RTT is requested to be reduced by accepting a downgrade in the available bandwidth as a counterpart. I find these calls interesting for services like online videogames or videoconferencing systems, among others. The contrary might be also interesting for other use cases and many other exchanges will be also inspected as part of my future work. The definition of an instantiation of this proposal and specifying in detail how it could be achieved by employing SDN is also part of my schedule. Finally, the biggest challenge is how to model a SLA with the client that would allow this kind of exchanges so that it can be enforced by ISPs, for that legal advice from researchers in legal areas will be pursued.

3 A Permissioned ISPs Blockchain For The End to End Quality of Service Enhancement

In this work I designed a permissioned ISP blockchain for the end to end (e2e) quality of service (QoS) enhancement that shall be framed in an Internet Service Provider (ISP) cooperative scenario. The designed blockchain is able to store users' connection data, in those connections where a lack of Quality of Service is experienced. Those data will be retrieved from the Networks Management System (NMS), and in some specific cases can be retrieved from the SDN controllers, of the ISPs chaining scenario that I recently proposed.

I designed the blockchain and I defined the smart contract by using Hyperledger Fabric. The need for a cooperation among ISPs, in order to provide the OTT's final users with a better QoS, has been highlighted in several works. Ahmad et al. [32] proposed a collaboration model where a strategy is described, to increase the revenue generation and QoE for the OTTs and the ISPs.

In my previous work [2] a mechanism to provide a communication among OTT and ISPs has been provided. REST APIs have been proposed in order to create a channel of communication among OTT and ISPs, with the purpose of improving the end to end QoS of the final users. In this work I designed a permissioned blockchain for an ISPs cooperative scenario where user connection data is stored. I chose to implement the designed blockchain by using Hyperledger Fabric [33], above all because it permits the use of channels and the creation of separate ledgers.

The paper is organized as follows. The design of the blockchain is provided in Section II. In section III some aspects are discussed related to the embryonic deployment of the Blockchain. Section IV draws the conclusions of this paper and the future direction of the research.

By using the method that I defined in my REST API, such as GET minmaxrtt, GET availablebw, POST rttreduction and POST exch-av-bw4rtt, it is possible for a certain OTT to retrieve information regarding the user's connection in a certain ISP's network. With the two GET methods proposed it is possible to know which is the minimum and maximum RTT, how much bandwidth is available for a certain user, depending on the initially contracted bandwidth, while with the two POST methods proposed, it is possible to ask a particular ISP to reduce the RTT for a certain user, or to exchange the eventually exceeded bandwidth with a RTT reduction.

By designing these methods, the problem of how to create a secure channel of communication arose. In particular, how the information transmitted via these REST APIs can be trusted by the participants to the communication. For this reason I decided to design a permissioned blockchain for the previously described ISPs cooperative scenario. I chose to try to implement the designed blockchain by using Hyperledger Fabric, above all because it permits the use of channels and the creation of separate ledgers.

Furthermore, the use of X.509 certificates, released by the Hyperledger Certification Authority server, guarantees a high throughput in terms of quantity of transactions supported, and at the same time a certain level of security of the blockchain network.

3.1 Design of the permissioned Blockchain

In my design I considered that certain data can be seen by all the participants, in order to provide the required improvement in the QoS delivered to the final users, but other data can be confidential and hence has to be exchanged just by the OTT and some specific ISPs. To do this, I used the channels approach architecture of Hyperledger Fabric. This approach makes use of established channels to a subset of participants in which only a determined set of transactions can be visualized, namely those related to the participants of a certain channel. This is thought as a network overlay, above the underlying blockchain.

My architecture is composed of a network consisting mainly of: clients, peers and ordering services nodes, smart contracts, channels, ledgers, and Fabric Certificate Authorities. Nodes, in general, are the entities that are communicating in the blockchain. They can be considered just as a logical function, being possible to run multiple instances of them in the same physical server. It is important to consider how they are grouped in trusted domains, and how they are associated to the logical entities that are able to control them.

The clients, also called submitting clients, are designed to be the representation of the end-users. They are in fact those who submit an actual transaction invocation to the endorsers, and those in charge of broadcasting transaction proposals to the ordering service. In order to interact with the blockchain they are connected to a peer node. The peer nodes are network entities owned and managed by the members of the blockchain, that maintain the state and a copy of the ledger, besides performing read/write operation. This particular kind of node receives ordered state updates in forms of blocks from the ordering service.

They also can play the special role of endorsers. This special function is performed towards a particular portion of a smart contract and consists in endorsing a transaction before it is submitted. Inside the smart contract an endorsement policy can be defined, which refers to a specific set of endorsing peers, where the necessary and sufficient conditions for a valid transaction endorsement are defined.

In the designed architecture the SDN Controllers are the peer nodes, to which it is also delegated the role of endorsers, being themselves the entities able to enforce policies inside their network. The ordering services are the ones in charge of providing a shared communication channel to peers and node, and are able to broadcast messages to all the peers, in the same logical order in which they have been output.

The access to the channels is permissioned, and they can be considered as a further private blockchain overlaid to the main blockchain, where data isolation and confidentiality are provided. Their definition is made by the configuration block, that resides in the channel specific ledger itself. The smart contracts, also called chaincode, are programmable containers where it is possible to define the functions that will permit the interaction with the ledgers.

The number of ledgers will vary depending on the number of the channels, considering one per channel. Every ledger is composed of the blockchain, where the immutable and sequenced records are stored in blocks, and the state database, where the current fabric state is maintained. The Hyperledger Fabric Certificate Authority (CA) issues the certificates that will authenticate OTT, ISPs and users to the network.

3.1.1 The Membership Operation Architecture

The access to the system is allowed through a trusted Membership Service Provider (MSP), whose task is to deploy a membership operation architecture, performing all the cryptographic mechanisms and protocols requested for the release and the validation of the certificates and the user authentication. Two or more nodes are able to create a channel, and only these nodes are able to access to the data transacted, thus providing a certain level of privacy and confidentiality that is bounded to the security of the channel. Upon every created channel, the participants will be able to create a separate ledger of transactions.

The importance of this feature is highlighted by the particular nature of the participants to my b , where all of them are potential competitors, but where their collaboration with the other entities involved is crucial for the delivery of a better QoS to the final users.

3.1.2 The Channel Configuration

Every channel contains its own shared ledger that has to be considered as an overlay used for data isolation and confidentiality. Only authenticated entities can access to the information transacted inside the channel. The configuration block defines the channel, and contains a single configuration.

When the configuration of the channel changes, a new configuration block will be forged, this process is called configuration transaction. The first of these blocks, where the initial configuration is used for the bootstrapping of the channel, is called Genesis block. The configuration blocks store data related to the organizations which are members of the channel and according to which channel access policies have to be applied. Also the block batch size is defined inside of it.

3.1.3 The Shared Ledger

The Hyperledger Fabric ledger subsystem is composed of two elements, named the world state and the transaction log. Every node owns a copy of the ledger to which it is belonging. The world state is the database of the ledger where it is described its state at a given point in time, while in the transaction log, all the transactions belonging to the current value of the world state will be recorded. It can be considered the update history registry for the world state. The whole ledger results to be a combination of these two components.

3.1.4 The Smart Contract

The smart contract that I defined, will be used by the External Service Application (ESA) [2] where my APIs will be implemented or by the SDN controller itself whenever the APIs will be integrated in its Northbound interface. By calling the functions defined in the smart contract, the exchanged information will be stored as a transaction into a block. In Hyperledger Fabric, nowadays the chaincode can be written in Go language or in Node.js. For my development I chose to use Go.

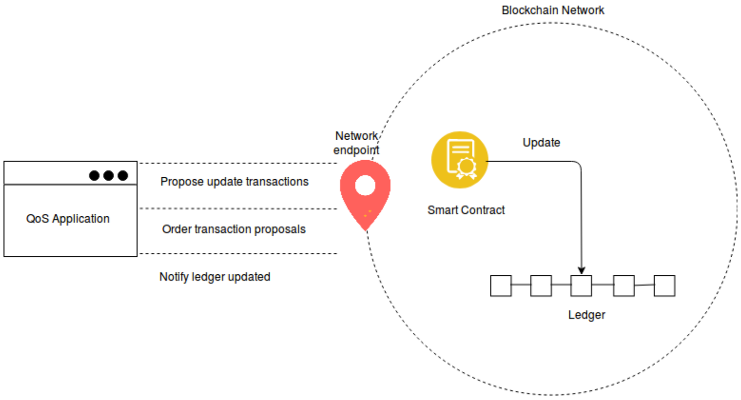

Figure 6: QoS triggering.

The smart contract that I designed will be used to store the responses to REST API calls shown in table 2, where four methods are described, which can be used by OTTs in order to retrieve information regarding the user connection in a certain ISP's network. I designed my smart contract to store in the ledger the responses to these calls.

The logic of functioning can be represented, as shown in figure 6, by this example: the call to the method *GET /qos/minmaxrtt?network = IPv4/netmask* is triggered by an OTT towards a certain ISP, asking for information regarding the minimum or the maximum RTT for a particular host, or network of hosts, whose traffic is crossing the ISP's network.

This call will be performed by an application that has to be developed with the Hyperledger Fabric SDK, that will interact via a network endpoint with the smart contract. The application will propose an update transaction. The smart contract through its functions will update the specific ledger for the channel. The access to this ledger will be granted at least to the OTT that is requiring the information and to the ISP that is giving back the response, in case the ISPs along the path among OTT and final user are not collaborating each other. In case they have stipulated an agreement, the channel will be accessed by more than one ISP, facilitating the communication among the peers and user's connection related information will be shared with all the participants.

In these specific cases, the ISP's ESAs, or their SDN Controller, will perform this action through an application that has to be written using one of the released Software Development Kits (SDKs). Nowadays, Hyperledger Fabric is offering two official SDKs, one for Node.js and another for Java, but unofficially, the SDKs for the languages Python and Go, also are available for the download and testing.

The development of this application will be covered in future works, together with a testbed deployment, where the implementation will be used in a simulated ISPs cooperative SDN scenario.

3.1.5 The Ordering Services

The ordering services are a defined collective of nodes that orders transactions in a block. The transactions are ordered basing on the first come first serve principle and are independent on the peer processes. Members are tied by cryptographic material which is contained in this element. Due to the heterogeneity of the participants, more CAs will be deployed, in order to provide a choice to the entities involved in the communication.

In my architecture the certificates are also used by the organizations to authenticate the transaction proposals generated by their applications. If I consider that the transactions committed are valid, the peers will use their certificates to endorse them. Ordering services are a crucial point in the Hyperledger Fabric architecture because the channel that they provide supports the atomic delivery of all the messages. This technique is also called total-order broadcast, atomic broadcast, or consensus.

Through the ordering service it is also possible to provide support for multiple channels, getting close to the messaging system of publishing and subscribe.

3.1.6 The Fabric Certification Authority (CA)

The Fabric CA provides a server and a client component. The server component is the one that releases the certificates, while the client one is used to register new identities and enrolling new peers. In figure 7 it is shown how the server fits in the overall architecture. In particular it is possible to interact with the server via the client component or via one of the provided SDKs. All the communications towards the server are implemented via REST APIs.

As shown in the figure, the architecture made use of a HA Proxy load balancer, to distribute the requests from a Fabric-CA intermediate server, towards a cluster of Fabric-CA servers. This latter will then communicate with the databases, among them it is possible to find the support for MySQL and PostgreSQL, but if LDAP is configured, it is possible to integrate it, and the identity information will be kept in the LDAP instead of in a database. Being my first implementation still in a very embryonal phase, I preferred to use only MySQL.

3.1.7 Specific Use Case: e2e QoS delivery enhancement.

The OTT requests a reduction of the RTT for a certain user. The OTT, in this specific use case, represents one of the client nodes that is submitting a transaction invocation to some endorsers.

The enhancement of the QoS for a particular final user is the object of the smart contract. This request will be endorsed by the peer nodes, that in this case are represented by the SDN controllers along the path. I assumed that all the ISPs along the path are implementing an SDN ruled by a controller, or a set of them, that is interacting with the permissioned ISP . In case the ISPs are members of the same channel, the OTT's request will be made just once.

In case the ISPs along the path are not all members of the same channel, the OTT request will be repeated more times, until the requested QoS for the final user will

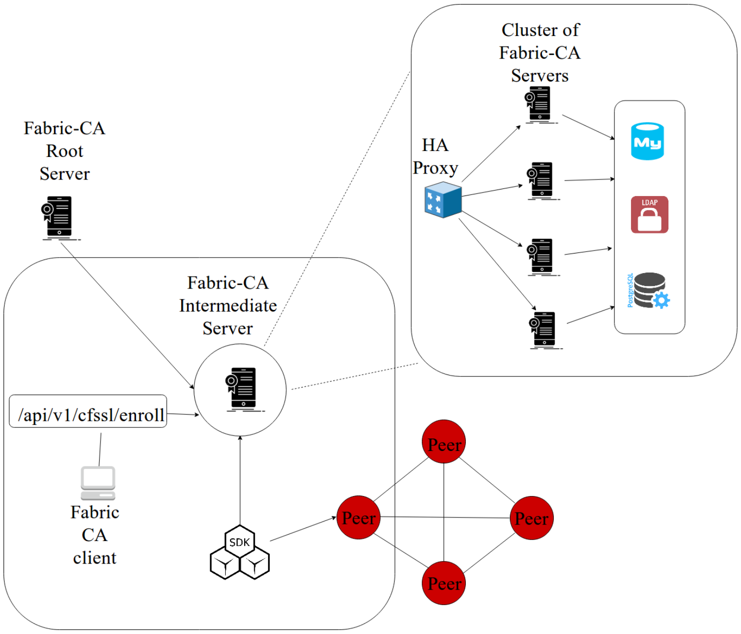

Figure 7: Fabric Certifaction Authority.

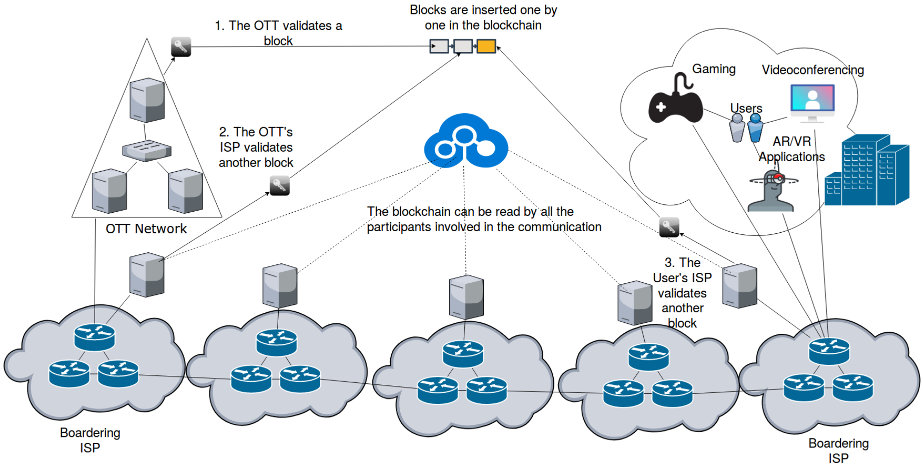
1. The OTT validates a block
Blocks are inserted one by one in the blockchain
Gaming
Videoconferencing
Users
AR/VR Applications
2. The OTT's ISP validates another block
OTT Network
The blockchain can be read by all the participants involved in the communication
3. The User's ISP validates another block
Boardering ISP
Boardering ISP

Figure 8: ISP simple scenario.

be achieved. It is obvious that, choosing a channel where the ISPs are connected together, namely where already the trust among them has been established, can be more profitable because the ISPs along the paths have more interest in delivering to the OTTs and the final users the QoS requested.

This approach brings advantages for everyone, from the OTT's point of view, the quality of the provisioning of its specific service is related to some QoS parameters that a specific channel may already have demonstrated to be able to deliver. This disruptive blockchain broadens the horizons for adopting new, specific and well defined policies when choosing routing paths to utilize in order to provide a good end to end (e2e) QoS for low latencies applications.

By nature, the Blockchain is used as a leading technology in cryptocurrencies, and the scenario of cooperation among ISPs is highly suitable to the definition of a new cryptocurrency, or to the use of an already existing cryptocurrency, through which it would be possible to bill feasible improvements in the QoS.

This last approach would lead to a further increase in the efficiency of the management of network resources, such as for example the bandwidth devoted to each user, optimizing its usage and increasing the routing efficiency for those applications which require a low latency. However, it is important to underline that the employment of the SDN controllers proves fundamental in this approach, inasmuch they have demonstrated to be able to adopt a reactive behavior with respect to the information encapsulated in every new block added to the blockchain.

3.2 Implementation of the Blockchain

3.2.1 The Membership Service Provider (MSP)

Each instance of a MSP has to be configured locally in each peer, orderer and on the channel where they are communicating, in order to enable the identity validation and the authentication, performed by the verification of the signature. The network is created starting from the Ordering service, in which the configuration of the channel is contained. Each channel is configured via policies and membership information, using X.509 root certificates.

A consortium of two or more organizations is defined by ISPs that need to coordinate their QoS parameters related to a certain user, agreeing on policies that rule the network. In the cooperative ISP scenario described in [2] a consortium is represented by the ISP chaining that connects the OTT to the final user. The policies will be applied by the SDN controllers of the ISP networks providing a Round Trip Time (RTT) within the desired thresholds, in order to provide a good QoS to the final user.

After the creation of the consortium, a channel is created, representing the communication layer that will be used to connect the entities involved in the communication.

3.2.2 The Channel Configuration

The channel configuration is done creating a configuration block that, being the first, is called the genesis block. In this block no other transactions will be stored.

The channel configuration owns three important properties: it is versioned, it is permissioned and it is hierarchical.

In case a new organization is added to a channel, the steps to follow are the following: to generate its crypto material by using the cryptogen tool, to prepare the command line interface (CLI) environment by running a docker instance and exporting the ORDERER_CA and CHANNEL NAME variables. Then a Protocol Buffer binary file will be saved, where the channel configuration block will be contained. This block is generated by fetching the configuration using the peer channel fetch command.

The previously created protobuf binary file containing the configuration block, has to be converted into a JSON format by using the configtxlator tool. Furthermore, the block will be cleaned from those headers, creator signatures and metadata that are not relevant to the change that has been done, by using the jq tool. Then the crypto material that had been generated before has to be added to the definition of the new organization configuration.

After this step, the configuration update has to be signed and submitted. Finally, before a new organization is joining the channel, the leader election process has to be configured. The last part of the joining process, is the updating and the invocation of the chaincode during which its new version will be installed.

The generation of the crypto material is done by means of the yaml files contained inside the artifacts directory of the new organization. Information such as name and domain name of the new organization is contained in the related YAML file.

3.2.3 The Smart Contract

The invocation of the function RTTreduction of the smart contract that I defined, containing the request to reduce an RTT for a certain user, is done as consequence of the validation of a transaction proposed by the OTT application towards a particular ISP, or group of ISPs. ISPs are able to reduce the RTT for a certain user giving priority to its flows, and adopting best path routing techniques. In my previous work, I also considered the possibility to exchange the exceeding bandwidth with a reduction of the RTT.

The lag experienced in certain real time communications is related to the RTT, and it is possible that the bandwidth contracted by the user with its own ISP is not totally used. The application of smart contracts to this situation will provide advantages to all the participants, namely, will give the OTT and the final user the desired QoS, while for the ISPs it will be possible to know exactly which amount of bandwidth a certain user is not using at all, and employ the unused one in other ways, for example improving the connections of those users that are not using low latencies services.

3.2.4 Future directions.

In the conclusion of this first step towards the full deployment of a permissioned ISP blockchain, I found several native features offered by Hyperledger Fabric very important for my scope. In particular, the possibility to create channels and to use a specific ledger for every channel created, made us further reflect about the need to create consortia of ISPs with the purpose of cooperating towards the same direction, namely the improvement of the QoS for final users.

The next step in my research is related to the deployment in a real testbed of a simulated ISPs network, where the ISPs cooperative scenario will be proposed. In order to reach that step I need to configure the SDN controller to be able to read inside every new block of the ledger, and to program them to have a reactive behavior depending on the information contained in the blocks. The usage of a blockchain in a chaotic scenario such as the ISPs network traffic, will permit to create a new generation of network administrators more and more oriented to the user's flows optimization, bringing this new generation to be skilled network software defined programmers.

A further step in my research is the design and the implementation of a permissioned ISP blockchain ICN oriented; namely, a hybrid ICN blockchain, able to exploit the native in-network cache and multicast functionalities offered by several ICN implementations, such as NDN and CICN, that are the focus of my studies.

4 An ICN Blockchain for the Quality of Services Improvements

In order to provide a better Quality of Service (QoS) to the final users of an Over The Top (OTT) service provider, a permissioned Information Centric Networking based blockchain is proposed in my work.

Through the Smart Contracts proposed here, improvements in the end to end connectivity for those OTT's users that are suffering a lack in the delivered QoS can be provided. In order to do this, the OTT is able to ask the Internet Service Providers (ISPs) along the path for some information regarding the user's connection, by calling the methods provided by the REST APIs.

When the retrieved values are under the desired thresholds, the execution of the Smart Contracts is performed between the OTT and the queried ISP. The designed blockchain can be used in an OTT-ISP cooperative scenario where the improvements in the user's connection can be billed by using the Ether cryptocurrency and a Proof of Authority (PoA) consensus algorithm is implemented.

Multimedia services such as videoconferencing, on line gaming, video streaming and augmented and virtual reality applications require more and more bandwidth. Users need to perceive a fair Quality of Experience (QoE) in order to remain loyal to their Over The Top service providers (OTTs). The perception of the QoE is doubtless influenced by several Quality of Services (QoS) parameters that can be measured in the Internet connections, such as for example latency.

In this specific case, increasing the provided bandwidth would not be enough to solve the lag problem in low latency services, which represents one of the most annoying issues. Instead of the bandwidth increasing, a reduction of the Round Trip Time (RTT) would bring the desired results.

Final users that are enjoying multimedia services, OTT service providers and Internet Service Providers (ISPs) along their path are the three main actors in this scenario. All of them would benefit from a cooperative scenario where some information about their connections is shared.

In some specific cases, the OTTs are interested in improving the end to end connectivity towards their final users when a bad QoS for their low latency services is suffered. To solve this problem, the need of a mechanism for the communication among OTT and ISPs has been highlighted. In order to perform such improvements, the ISPs involved will be those that reside along the path between the users and OTTs.

In order to raise the standard of QoS parameters, in an ISPs cooperative scenario I proposed in [2] REST APIs for the communication among OTT and ISPs. With these REST APIs it is possible to retrieve user's connection QoS parameters, such as the bandwidth and the minimum and maximum RTT.

In [1] I highlighted the ICN paradigm as a possible candidate to solve the QoS problems affecting real time multimedia applications. ICN enhances content delivery using features that the current IP network is not offering, such as data caching and the request or the route of content at network layer, using the name of the content instead of the host's address. It also provides transparent caching to final users, which are agnostic of the localization of the content. This approach reduces latency in the content access improving the network efficiency.

Furthermore, ICN offers native multicast transmissions. In traditional IP networks it is usually used for streaming media and other network applications, but often ISPs block the reserved multicast address because they use it for their network operations. Multicast over ICN solves this problem. The adoption of multicast transmissions reduces the resources consumed by a server that is providing the same content to more users.

Even though hybrid versions of ICN could be used by OTTs for the provisioning of real time services to final users, in this work I am proposing only the design of an ICN permissioned blockchain for the enhancement of the QoS for the final users of low latency services. The designed blockchain is used to store the information related to the user's connection, gathered by the invocation of the methods provided in the REST APIs. My design is considered to be a sidechain of Ethereum.

I also defined Smart Contracts (SCs) where actions aimed at QoS improvements billing services are defined. These SCs will be executed when the retrieved parameters are below the desired thresholds. I adopt the Proof of Authority (PoA) as blockchain consensus algorithm. The adoption of PoA increases the performances in terms of transaction throughput. The OTT plays the role of authority node, while the bordering ISPs play the role of validators.

4.1 Related Works

In [34] Lavbič et al., introduce a novel architecture for monetization using Smart Contracts (SCs). Its goal is to improve the QoS when utilizing their home made video-conferencing (VC) system in business context. They implemented their SCs relying on the Ethereum network, writing them using the Turing-complete programming language Solidity. Also they have shown in their study that the Ethereum cryptocurrency represents the lowest cost in transaction processing fees with respect to the traditional methods of monetization, such as Visa, Mastercard and Paypal.

They packed the components forming the proposed VC architecture into a Docker CI, and by using the orchestrator proposed in [35] they addressed the needs to provide a flexible usage of the resources requested by the VC service. Such usage can be measured and billed through the implemented blockchain technology.

In their architecture the users interact with the blockchain layer via a web GUI, that enables the execution of the SCs and the interaction with the distributed ledger. In turn, the blockchain communicates with the VC service, which by a monitoring system and the orchestrator asks the cloud providers for the required amount of software and resources.

In [36] Sedky et al. proposed Blockchain-Centric Exchange Protocol (BCXP), which is an information-centric architecture used to optimize information exchange. Their proposed protocol resides on top of a three layer architecture. At the bottom there is the *NDN interest/data* layer while in the middle there is the *COPSS publish/subscribe* layer. With respect to BlockNDN in their work they implemented a publish/subscribe mechanism which is originally missing in the NDN architecture.

4.2 Problem description: the Blockchain scalability in traditional networks

In Blockchain systems integrity is maintained via consensus reached by network peers. Such integrity is affected by several parameters, such as inter-block time and the number of transactions per block. The achievement of the consensus is also influenced by network bandwidth, peer-to-peer network and block propagation delay.

In bitcoin network, for example, the block propagation delay is mainly due to the traffic redundancy generated by the separation from the physical to the logical topology of the network. Data transmission becomes more complex as the number of peers increases, causing a bottleneck in the blockchain's transfer layer that represents an issue for scalability [37].

Regarding Ethereum scalability Wood, in [38], announces that it will remain an eternal concern, unless some form of hierarchical structure is adopted, such as, for example, the consolidation of lighter-weight chains within the main block or the construction of the main block by combining and adhering incrementally smaller transaction sets, allowing parallelisation of transaction combination and block-building.

Several approaches to solve this issue have been already provided in literature. One is the increment of the block size that is related to the network's bandwidth while propagation delay is increasing. This approach brings to smaller blocks the advantage of being committed by the majority of peers.

Another approach is to raise the degree of peer connectivity, which is bound to the increment of the network overhead [39] [40].

4.2.1 Existing ICN approaches

In works like BlockNDN [37] and BCXP [36] it has been shown how the exchange of information has been optimized deploying blockchains over NDN architectures.

BlockNDN is a bitcoin-like blockchain system and has been implemented and deployed in a cluster. The authors claim that serious problems that are affecting real blockchain systems in IP network are the lack of support for multicast and the hierarchies of status. In their design those problems have been solved. BlockNDN provides completely decentralized systems and simplifies a system architecture. Furthermore, weak-connectivity phenomenon has been improved and the broadcast overhead has been decreased.

BlockNDN uses directly the physical topology to broadcast transmissions. The nodes just request data by sending interest packets. These packets are forwarded along the physical paths by the neighbors, until it is discovered who has the requested data. Whenever a node is able to send back the data requested, data packets will be forwarded to the reverse path where the interest packet has been forwarded.

The efficiency in the transmission of the Blockchain, increasing the network performance, is given by several factors. Broadcast overhead is reduced thanks also to the in-cache functionalities provided by the NDN routers, that are equipped with Content Store (CS). Native multicast is able to decrease the complexity of the data exchanged. When a new block is generated a multicast transmission occurs. If Alice, for example, caused a state change, Alice's new data will be multicasted to the other users, following the Packet Interest Table (PIT) entries set up in routers by sync interests.

TCP/IP protocol does not provide any native support for multicasting. In order to permit multicast traffic, extra work should be done. BlockNDN uses ChronoSync [41] over NDN as synchronization protocol.

On the other hand, in BCXP [36] a customized blockchain protocol on top of NDN has been implemented. BCXP implementation overcomes the network partition problem that affects ChronoSync. The BCXP architecture is composed of three layers. At the bottom there is the NDN interest/data layer, on top of this the COPSS publish/subscribe layer, while the upper one has the BCXP Namespace. The adoption of an extended version of Content Oriented Publish/Subscribe System (COPSS) [42] makes it possible to adopt a publish/subscribe model, a communication pattern which is not natively supported by NDN.

Using NDN interest/data model is sufficient for pull-based applications, where data transfer is initiated by the receiver. However, this is not the case in blockchain networks, where receivers are not aware of blocks/transactions generation time. This use case fits perfectly with a publish/subscribe model. Here, receivers should subscribe to namespaces they are interested in and, as soon as blocks/transactions are produced, they are broadcast to all subscribers.

4.3 A permissioned blockchain design

In the following analysis I focused my attention on how Ethereum implemented its consensus algorithm, because my final goal is the deployment of my ICN blockchain as an Ethereum sidechain. My main idea is to adopt Ether as cryptocurrency, providing an implementation of my blockchain where the proposed Smart Contracts can be run.

4.3.1 An analysis of consensus algorithms adopted in blockchain.

Ethereum [38] can be viewed as transaction-based state machine. It started with a genesis state and then it executes transactions incrementally turning the machine into some final state. That final state is the canonical "version" of the world of Ethereum. A blockchain is a collection of blocks linked to each other where transactions are embedded. In each block there is a header, where meta information is stored, and which contains an ordered series of transactions.

Blocks are created by entities called miners, through the mining process. This process has the same name used in mines, where gold is extracted, because the discovering process of the digital currency can be as difficult as gold mining. In the early history of Ethereum the mining process was carried on by using a custom version of the Proof of Work (PoW). The PoW is the original consensus algorithm used in blockchain network, and consists in the resolution of a difficult puzzle. The concept upon which PoW is based was originally proposed by Dwork et al. in [43] in 1993, while the term was coined and formalized in 1999 in [44] by Jakobsson and Juels.

The first implementation of PoW was introduced in 2002 in Hashcash [45] when it was improved and experimented by using its CPU cost-function to compute a token that can be used as PoW. Finally, in 2009 the bitcoin network [46] went on-line adopting the PoW as consensus algorithm, and pushing this system to be the most secure and widely adopted in cryptocurrencies.

However, the huge amount of electricity required to run the PoW, brought the Ethereum developers to announce that their platform will switch to a pure Proof of Stake (PoS) blockchain, when the Serenity version is released. The proprietary algorithm run on Ethereum is called Ethash, and its execution is done by giving as input the block header of the last accepted block, in which a randomly generated number called nonce is included. In output Ethash returns a hexadecimal number. By modifying the nonce, the output given by Ethash changes. The header of the block that the algorithm returns must be less than the network difficulty, which represents another hexadecimal number which plays the role of target to be beaten [47].

The resolution of the Ethereum PoW is easier than the one adopted in the Bitcoin network, in fact a new block is discovered every 15-30 seconds, against the 9 minutes required to mine a new block in the latter. Whenever a node mines a new block, it receives a reward. In both systems the network difficulty is incremented, in order to maintain the time required to mine a new block constantly. During the Ethereum mining process two miners might produce a block around the same time, but only one can be accepted in the main chain. In this specific case, the unaccepted block is also included in the chain, even though it will receive a minor reward, taking the name of uncle block.

The rules that are used in the validation process of a block by the miners, may change among different versions of the client used to interact with the blockchain.

In fact, by updating the official client, which for the Ethereum platform is called Geth, also the set of consensus rules may change. When the miners are running different versions of this software it is possible that they manifest their consensus using different subsets of rules. In this specific case some blocks will not be rejected by those clients that are running the new rules, thus making it possible that soft forks of the blockchain occur.

When a software update presents a totally different set of rules with respect to the old one, and a group of miners do not update their software, hard forks may occur, bringing to a split of the chain. In this scenario some blocks may be valid for one chain but not valid for another. Ethereum platform has been hard forked already six times.

4.3.2 A Proof of Authority Blockchain design

The choice to adopt a permissioned blockchain is due to the major throughput that can be achieved in number of transactions per seconds. The limit in the number of transactions in the first and second generation of blockchain, such as Bitcoin and Ethereum, is in fact a well known problem. This limit is considered to be the main reason for the slowdown in the adoption of these cryptocurrencies for the purchase of goods in everyday life.

The limit is due to two main factors, the time required for the updates propagation of the blockchain among all the peers, and the amount of transactions that can be written inside each new forged block. A possible solution to overcome this limitation is to augment the size of the block, thus permitting to insert more transactions. On the other hand increasing the block size will further reduce the broadcasting speed of the updates.

These reasons led us to avoid the use of a permissionless blockchain. Other approaches, different from the one that I adopted, can be considered, such as the use of Lightning networks [48]. This approach enables fast transactions, thanks to the adoption of a second layer that operates on top of the blockchain where a payment protocol is executed. The protocol avoids broadcasting the transactions by opening a payment channel, where two nodes can make use of Smart Contracts.

My approach is based on a permissioned implementation of Ethereum, exploiting the use of Solidity as programming language and the Ethereum Virtual Machine (EVM), combined with the blockchain, in order to execute SCs. This approach represents a good solution, in which the usage of Ether as cryptocurrency is permitted in order to make the billing of the service improvement, while the access to the blockchain remains restricted.

I chose to design my SCs using an Ethereum based platform called POA Network, which is a sidechain to Ethereum that uses the Proof of Authority (PoA) as its consensus algorithm. Thanks to the adoption of PoA the transactions can be delivered instantly together with a coherent consensus. PoA is a consensus algorithm introduced by VIVA, more efficient than Proof of Work (PoW), because it does not depend on energy consuming mining farms, and more effective than Proof of Stake (PoS), because of the avoidance of top-heavy stake holders dominating the whole ecosystem.

VIVA forked Hyperledger Fabric creating an Ultraledger multidimensional blockchain. In PoS system, PoW issues such as the quantity of electricity required or the 51% attacks, are solved by adopting the usage of validators and avoiding the resolution of a puzzle in order to generate new blocks. Those who want to perform the role of validators have to deposit a certain amount of Ether in a Smart Contract to participate in the consensus system.

For example, in Casper version 2, which is the successor of Casper Friendly Finality Gadget (FFG) [49], a hybrid PoW/PoS system, the amount of Ether that has to be deposited in order to be a participant in the Casper consensus system is 32. A limit represented by this approach is that a node can be a validator just by depositing a certain amount of a cryptocurrency, and the benevolent behavior of the node depends just on the interest that it has in the amount of money that it deposited. Namely, if a validator behaves in a malevolent way, the network will simply lock this Ether away. The PoA consensus mechanism has also been implemented by Kovan and Rinkeby in their respective Ethereum testnets.

My adoption of the PoA is confined to a private network usage, but it is also suitable for public networks, where trust is distributed. In PoA, such as in PoS, validators are in charge to forge new blocks, but the approach taken to elect them is slightly different. In PoA validators are called also sealers and they perform the role of authority nodes. A certain number of them are pre-approved and when a new node wants to join, the sealers pool needs to be voted on by the pool itself. In this way it is possible to control the sealers that are mining blocks in my blockchain.

Validators have to be formally verified on-chain via Decentralized Applications (DApps), their identity information has to be available in the public domain in order to be cross-referenced by every one and only one identity per person is allowed. Depending on the number of validators N, in order to avoid that a malicious one could severely damage the network, each validator can sign at most one of a number of (N/2) + 1 consecutive blocks. Until the private key of a validator is not compromised, the security of the blockchain is not undermined.

In my architecture the pre-approved authority nodes are represented by the OTT and the bordering ISPs, namely the one that is connecting the OTT and the one that is connecting the final users. The ISPs along the path will be automatically inserted in the sealers pool if they are connecting other OTT's users. The designed blockchain is intended to satisfy the users of a specific OTT, but more blockchains can be deployed, involving users that are enjoying more OTT's services. In figure 9 I showed the basic scenario where an OTT is performing his role of validator, together with the two bordering ISPs, while a user is enjoying the OTT's service.

4.4 A Smart Contract for the gaming delay use case

In the scenario described in figure 1 a user is enjoying a service from an OTT, by being connected to its ISP. The user's connection passes through other three ISPs and finally reaches the OTT's ISP. By using the REST APIs in table 2 that I proposed in [2], the OTT is able to know if and where the user's connection is suffering bad QoS. For example, if the user is playing a game that needs at maximum 30 ms of delay towards the OTT network, and assuming the distances among ISPs are equally

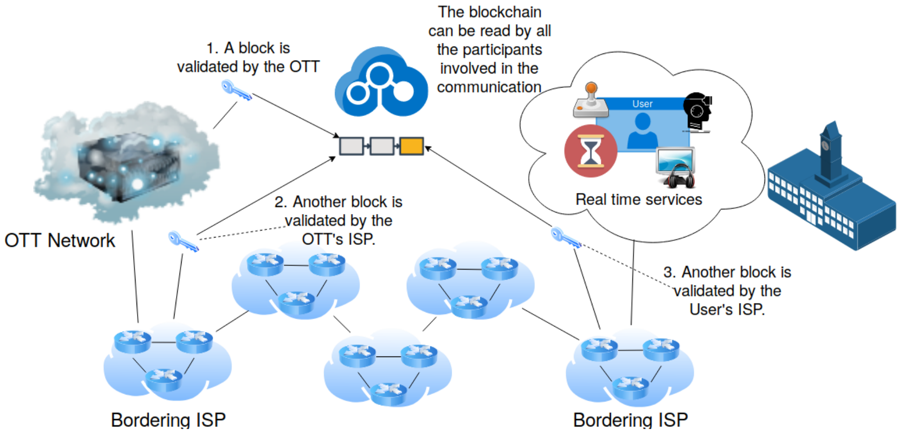
1. A block is validated by the OTT
The blockchain can be read by all the participants involved in the communication
User
Real time services
OTT Network
2. Another block is validated by the OTT's ISP.
3. Another block is validated by the User's ISP.
Bordering ISP
Bordering ISP

Figure 9: ISP blockchain simple scenario.

weighted, the maximum delay supported among each ISP will be 6 ms, considering also the same value for the OTT and user connection with their relative ISPs.

A pseudo code for the definition of the Smart Contract that can be used for the described scenario can be the one showed in the pseudo code below.

```
pragma solidity ^0.4.22;
//@title Gaming delay
contract GamingDelay {
    /**total delay among
    OTT and its users*/
    uint MaximumDelay = 30;
      uint MaxSegmentDelay = MaximumDelay/(ISPsQuantity);
    uint DelayReducted;
    //Maximum round trip time
    uint maxrtt;
    /**Number of ISPs
    that are in the path*/
    uint ISPsQuantity = 5;
    /*Number of Ether transferred for a
    certain QoS improvement*/
    uint EtherTransferred;
    //If true,the delay is ok
    bool DelayOk;
    bytes32 IPAddressISP;
    address OTTWalletAddress;
    address [] ISPsWalletAddress;
    /**Array containing the users
    IP addresses*/
    bytes32 [] UsersIPAddress ;
    /**Array containing the ISPs
    IP addresses*/
    bytes32 [] ISPsIPAddress ;
    /**Array containing those ISPs
    that are not able to provide
    improvement in their connections
    when parameters are below the
    desired thresholds.*/
    bytes32 [] ISPsBadQoS;

    function CheckDelay(int n) {
        for ISPsIPAddress[i]
            execute method GET
            /qos/minmaxrtt?network = UserIPAddress;
            return maxrtt;}
    function bool CheckThresholdDelay(){
        for (uint i=0; i<ISPsQuantity; i++){
            maxrtt= CheckDelay(i);
            IPAddressISP = ISPsIPAddress[i];
            if ((maxrtt/2) > MaxSegmentDelay)
                { DelayOk = false;
                IPAddressISP=ISPsBadQoS[i];}
            else
                { DelayOk = true; }
            return DelayOk;}}
    function bool RequestDelayReduction(){
        DelayReducted = POST /qos/rttreduction?network=
            UserIPAddress/netmask&maxrtt=MaxSegmentDelay*2;
        if(DelayReducted <= MaxSegmentDelay)
            { PayForDelayReduction();
            DelayOk = true; }
            else
            { DelayOk = false; }
            return DelayOk;}
    function PayForDelayReduction(){
        from OTTWalletAddress send money to ISPsWalletAddress[i];
        return EtherTransferred;}}
```

In this Smart Contract four main functions are defined: *CheckDelay()*, *CheckThresholdDelay()*, *RequestDelayReduction()* and *PayForDelayReduction()*. The *CheckDelay()* function uses the GET method to retrieve the minimum and maximum RTT for the user's IP address, returning the maximum RTT. The *CheckThresholdDelay()* is a boolean function that returns true if the delay of the connection crossing a certain ISP is under the desired threshold, false otherwise.

The *RequestDelayReduction()* function invokes the POST method used to ask a certain ISP to reduce the RTT for the specified user to the indicated value. The accepted threshold expressed in ms for crossing each ISP network is contained in the variable MaxSegmentDelay. The *PayForDelayReduction()* function is the one used to make the payments from the OTT's wallet to a certain ISP's wallet. A possible execution of this contract can be the one shown in the pseudo code below.

```
for (uint i=0; i<UsersIPAddress.length(); i++){
    DelayOk = CheckThresholdDelay();
    if (DelayOk==false)
    {RequestDelayReduction();
    if (DelayReducted<MaxSegmentDelay)
    {PayForDelayReduction();
    if (DelayOk==true){
    WriteToNewBlock("QoS parameter improved for the user with IP
        Address:"+UsersIPAddress[i]+"which was suffering a lack of
        QoS while crossing the ISP with IP Address:"+IPAddressISP+"
        .The amount of transferred Ether from the wallet address:"+
        OTTWalletAddress+",belonging to the OTT, to the wallet
        address" +ISPsWalletAddress[IPAddressISP]+",is "+
        EtherTransferred+".");}}}}
```

In this execution the OTT is going to perform a check for the users IP addresses, by scanning the array containing them, and running the functions described above. If the PayForDelayReduction() boolean function returns a true value, it means that a transaction between the OTT and the ISP where the lack of QoS was detected, occurred. Finally this transaction will be written in a new block of the blockchain and validated by the sealers pool.

4.5 Future directions

In this work I presented a design of a permissioned ICN blockchain, defining Smart Contracts able to improve a specific QoS parameter, the RTT. This work is framed within the ISP-OTT cooperative scenario that I am currently investigating. I am now focusing on the implementation of the proposed blockchain in NDN.

I am also looking forward to improving my design by integrating the interaction with the SDN controllers of the ISP's network and OTTs, in a hybrid ICN fashion. My final objective is the proposal of a hybrid ICN model where real time multimedia services are managed cooperatively by OTTs and ISPs. The model will adopt a blockchain which will be used to bill QoS improvements via cryptocurrency.

5 A study on how Blockchain can improve the Management of Next Generation Networks

Internet traffic is growing together with the amount of connected devices and the Internet's best effort delivery model is not able to support efficiently such increase. The resource allocation problem generated by this change has been faced by means of design approach such as the Recursive InterNetwork Architecture (RINA) and by creating new network paradigms, such as SDN and ICN.

Recently hybrid approaches have been experimented in order to put together the benefits offered by different next generation network implementations. On the other hand, the emerging blockchain technology is able to provide these possible solutions with a method for authentication and data integrity, permitting the creation of a communication layer, where information useful for a better network management can be exchanged.

Furthermore, the development of a blockchain on next generation networks, such as ICN, will take advantages that allow to overcome well known issues such as low throughput in terms of transactions per seconds. According to a Cisco forecast, Internet video streaming and downloads are starting to take more and more bandwidth and will grow overcoming 81 percent of all consumer Internet traffic by 2021.

Most of this traffic is usually managed by the content delivery networks (CDNs), which exploit servers close to the network edge. CDNs' main task is to reduce latency and core network traffic. Notwithstanding CDNs' scalability is widely demonstrated, the scalability of the whole Internet is undermined by the rapid increase of video traffic together with the proliferation of Internet of Things (IoTs) devices.

By definition, inside an IoT domain every node can be a provider, bringing to more and more many-to-many communications, which in turn increase the size of the routing tables. The problem of resource allocation is due to the fact that the Internet's best-effort delivery model is not able to effectively respond to the new requirements.

This problem is affecting mostly Internet Service Providers (ISPs) and Over The Top (OTTs) while they are trying to deliver a good Quality of Service (QoS) to their final users. In order to tackle this problem, theoretical models such as the Recursive InterNetwork Architecture (RINA) have been proposed.

However, a concrete solution can be figured out by adopting the combination of next generation network architectures. Among these architectures there are two important network paradigms: SDN and ICN.

RINA is a network concept which lays on the Inter-Process Communication (IPC) model. Day et al. in [50] proposed a clean slate approach, which is based on the principle that application processes communicate through a distributed inter-process communication (IPC) facility. The application processes that compose this facility provide in turn a protocol which is implementing an IPC mechanism, and a protocol that is used for the managing of the IPC, performing routing, security and other management tasks.

Among the features itemized in their work, a very basic premise where a different vision with respect to the layered networking is described. In the IPC model the networking is conceived as a single layer of distributed IPC which is repeated for different scopes, in contrast with the layered networking approach, where each layer performs different functions. Among the next generation network architectures SDN is one of the most used in real scenarios.

By introducing a separation among data plane and control plane with SDN it is possible to provide better scalability and high availability with respect to the traditional IP networks. Furthermore, the monitoring functions of the network are improved by using the controller which is providing a global view of the network resources.

The ICN paradigm brings to a substantial change in the way in which information is exchanged compared to the traditional IP networks [51]. This change consists in the transformation of the traffic from host-centric to information-centric. The change is supported by in-network caching capabilities provided by the nodes and the infrastructure itself.

The goal of this approach is the provision of a network infrastructure more resilient to disruptions and failures, more suitable for today's use, with a special eye to content distribution and mobility. Several hybrid approaches of ICN overlaid on the traditional network have been proposed, among them it is important to mention virtualized ICN [52] that makes use in its experimentation of the SDN controller OpenDaylight.

Blockchain technology is an emerging technology whose main purpose is the provision of a shared ledger among its nodes enabling authentication mechanisms and data integrity. The Blockchain can be mostly of two types: permissionless or permissioned. In the first one, generally, a peer to peer approach is adopted and a distributed consensus algorithm is run in order to forge the blocks that compose it, and the participation is allowed to every user.

To preserve its security time or energy consuming algorithms have to be run, slowing down the throughput in terms of number of transactions per seconds. The permissioned one is restricted to only those entities that receive an invitation or ask to participate. Generally the consensus algorithm, even though distributed, is less energy consuming than the permissionless ones, because the whole security is mitigated by the imposed access restriction.

In the latter the throughput can be higher thanks to the less time elapsed in forging blocks. In this work I analyze how the blockchain technology and the next generation network architectures may both gain by their commutative adoption. In particular the first one will be speeded up by being run on the latter, and the latter may benefit from the adoption of a blockchain for the implementation of a safe and distributed authentication method also able to provide data integrity.

5.1 Related Works

As shown in the work DistBlockNet [53], where a distributed secure SDN architecture for IoT using the Blockchain technique is proposed, SDN controllers are able to have a reactive behavior, taking information directly from the Blockchain. In [53], only SDN and Blockchain are combined in order to try to satisfy the rising requests of the current IoT networks.

In fact smart devices need a more adaptive, flexible, efficient, secure and scalable architecture able to optimize the data plane of each local network participating in a distributed trusted wide area network. In their work, the authors try to satisfy these requests by proposing a distributed Blockchain, deployed in testbed with 6 SDN controllers. They showed, in their results, how a blockchain can be helpful in order to update the flow tables of each network, programming the SDN controllers to read the data contained inside the Blockchain.

The final goal of the proposed model is to provide protections, including threat prevention, data protection, access control, mitigate network attacks such as cache poisoning/ARP spoofing, DDos/Dos attacks, and to detect security threats. Being a reactive solution, also the attack window time is reduced thanks to a continuous update of the table of flow rules.

They evaluate the performance of the proposed solution taking into consideration several aspects: scalability, defense effects, accuracy, overhead analysis. For the scalability analysis, they used 6 servers Intel i7 with clock of 3.40GHz and 16GB of RAM, implementing a distributed blockchain network working with 6 controllers/verifications and 6000 request/response nodes.

Due to scalability reasons, they used OpenFlow software instead of OpenVSwitch, that does not scale well when a large number of switches are emulated. For the comparison, they also deployed a normally distributed SDN network. Their results showed that the flow table updates time with respect to the packet-in arrival rate, the proposed model constantly performed superiorly to the distributed SDN network, as the rate of the packet-in arrival increased.

For the defense effects analysis they deployed their testbed using two different test environments, one software-based and the other one hardware. In particular they used Mininet as SDN emulation tool for the software environment. In the hardware environment the python-based POX SDN controllers, OpenFlow switch and server machines implementing clients and data plane caches have been used.

Then they evaluated the validity of their DistBlockNet model, in both environments, measuring the bandwidth of the clients with and without flooding attacks generated by other clients towards the switch, generating these attacks at different speeds. They made an evaluation of the impact on the bandwidth with and without their solution, in both environments.

In the work BlockNDN [37] a blockchain has been designed and deployed, which uses directly the physical topology to broadcast transmissions. Jin et al. have shown that the ICN paradigm offers a better efficiency in terms of the propagation of the Blockchain. This is due to the possibility to implement the broadcast transmissions directly over the physical topology, avoiding dealing with the logical topology that is commonly used in the IP networks.

In [37] the efficiency in the transmission of the Blockchain, increasing the network performance, is given by several factors, in particular the broadcast overhead is reduced thanks also to the in-cache functionalities provided by the NDN routers, that are equipped with Content Store. Also it is important to consider how native multicast is able to decrease the complexity of the data exchanged. TCP/IP protocol does not provide any native support for multicasting. In order to permit multicast traffic, extra work should be done.

Compared to the Bitcoin network, where inv messages are used to avoid sending block messages to those nodes which have already received it, in the NDN the interest packet itself indicates the sender does not have the data in his Blockchain. This reduces the exchanging complexity and improves the propagation method of the system. In fact in their approach it is possible for a node to ask through an Interest packet only for a certain part of the chain.

The Interest packet sent out by every node which needs updates, will contain the hash of the last block that has been received. This approach represents a quick way to obtain missing blocks in case of disconnection of a node from the network.

In [54] Sharma et al. proposed a novel blockchain-based distributed cloud architecture, with SDN enable controller fog nodes at the edge of the network, in order to meet the required design principles demanded by the recent expansion of the IoT devices and the consequently explosion of data produced.

They adopted as main technologies fog computing, SDN and blockchain. Their proposed model wants to open new markets to the services offered through fog computing. The blockchain that they implemented brings together data from consumers and producers. Their model consists in four steps: selection of the resource providers, provision of services, registration of transactions and payment.

During the selection of the resource providers the cloud user chooses one of the providers present in the provider pool in the blockchain-based distributed cloud.

Once that a provider has been chosen the procedure will go on with the provisioning of the required service, that can range from task execution to data management, including the deployment of servers to the user.

After that the service is provided, the service provider will register the transaction in the blockchain and it will be shared with all the distributed peer service providers. In the last step the user will pay for the service enjoyed. The model proposed is able to put in communication, by using a blockchain, users that need some particular resource services and providers that are able to provide it.

The blockchain is also used to create a sort of feedback mechanism for the services provided, because the amount of resources required and the price paid for them will be transparent to all the participants. The blockchain that they designed is based on a consensus protocol that has been called Proof of Service, where the contribution of a certain peer is measured by some actions that occur outside the blockchain, such as the transferring of a file, or the performance of a computation, or the provision of a set of data.

These contributions will create the assumptions that will lead to the token exchange among members. Their architecture has been evaluated and its performance has been compared with other existing models. In their results they have shown that their architecture can significantly reduce the end-to-end delay between IoT devices, traffic load compared to the traditional IoT architectures and computing resources.

In [52], Sardara et al. contribute vICN that is a unified open-source framework for the network configuration and management which makes use of the recent progresses in techniques of virtualization and resource isolation. Most of the testbeds adopted by the research community for ICN experimentation are small scale or application-specific environments, not reusable in different contexts nor in real world IP deployments.

The proposed framework offers a single, flexible and scalable platform able to provide reproducible large-scale research experimentation, demonstrations made with physical and/or emulated devices and the required network resources to permit the deployment of ICN in the existing IP networks.

5.2 Network Architectures for a blockchain application

In this chapter I am going to drive an analysis upon model, network paradigms and technologies proposed in literature and in real case scenarios, to tackle the problem of resource provisioning that the current Internet architecture is facing.

5.2.1 Recursive InterNetwork Architecture

Day et al. with RINA [50] want to face the problem of resource allocation in their architecture by proposing an IPC model. In their proposed model a single layer can be duplicated repeating the same functions, but in order to operate on several ranges of the performance space, policies have to be tuned, such as for example delay, capacity and loss.

By adopting this vision, the greater the demand for resources by the network, the greater the volume IPC layers deployed, allows the architecture not to have logical limitation to its scalability. In RINA security is improved by avoiding an application to know any IP address or port of the connecting application.

In fact, in order to join a distributed IPC facility it is required an authentication ruled by policies, that can be different for every facility. IPC layers are stacked upon each other, and the network has been built with smaller and more manageable layers, in order to allow more granularity. This strategy supplies the providers with a better resource management, enabling them to adopt a divide-and-conquer approach.

Not only the basic networking functions that need to be provided as fundamental service are supported by RINA, but also its distributed IPC facility is able to support services, ranging from application relays, such as mail distribution, to peer to peer and transaction processing. RINA has been designed in order to remove the barriers created by the Transport Layer implemented in the current Internet, permitting ISPs to open new markets by the provision of IPC services directly to their consumers, thanks to the exploitation of their specialization in the resources management of lower layers.

Another strength which increases its scalability is represented by the use of private addresses to identify IPC processes inside each IPC facility. This allows its easy adoption in private networks, avoiding the tyranny of the current Internet architecture, where the amount of public IP addresses is limited.

5.2.2 Software Defined Networks

The term SDN is used to refer to a programmable network paradigm [55]. SDN is the ability of software applications to interact with the programming of individual network devices and therefore obtaining the control of the whole network [56].

Greenberg et al. in [57] described the main difference with the traditional networking paradigm. In SDN an abstraction between the traditional forwarding and the control planes has been introduced. This separation has been created in order to maintain the control of the forwarded data, divided from the decisions related to the control of the whole network. The main objective of this separation is to program distinctly the two planes, enabling faster innovation in both planes, while maintaining lower complexity in their development.

In the RFC 7426 [58] the following definition for SDN has been provided: a programmable networks approach that supports the separation of control and forwarding planes via standardized interfaces. The authors provided also a graphical abstraction of the SDN architecture in the form of high level schematic. That abstraction is represented and a hierarchical model in which layers can be stacked on top of each other and employed recursively, as needed, is shown. From the recursive point of view this approach is similar to the one adopted by RINA.

Programmability, faster innovation and easier management have been enabled by introducing a central entity called Controller, which is the one in charge to directly manage the control plane. Through the controller external applications are able to program the network [59], for example interacting with its northbound interfaces where specific APIs can be deployed.

Like for every architecture or system that relies on a central entity, issues related to scalability and reliability rise. The usage of the SDN paradigm spreads thanks to the adoption of vendor neutral and open control-data plane interfaces such as OpenFlow [55].

5.2.3 Multi-control approach in SDN architecture

Its open nature allows network hardware and software to evolve independently easing the replacement of proprietary hardware and firmware with commodity hardware and a free open source Network Operating System (NOS). In [60] an SDN distributed control platform called Open Network Operating System (ONOS) has been described and evaluated. ONOS is logically centralized even though its deployment is distributed across multiple servers.

It follows the footsteps of a closed source SDN controller called ONIX [61] which has been the first SDN controller to implement a global network view. Other two leading Open Source SDN controllers are Floodlight and OpenDaylight, both are able to run leader election process in a distributed system with multiple instances.

Hu et al., presented a comprehensive survey for multi-controller research in SDN [62]. While the network size increases, a single centralized controller is not able to satisfy the increasing demand for flow processing, causing its failure. To tackle this problem a multi-controller approach has to be considered. Several approaches have been analyzed and summarized in their work, but mainly the challenges that every approach has to face are scalability, consistency, reliability and load balancing.

Firstly they introduced two basic multi-controller architectures, which they concluded to be represented by the flat and the hierarchical design.

In figure 11 the Multi-controller flat design is shown. The controllers interact each other by using their east-westbound interfaces in order to get the global view of the network. A particular example of this kind of implementation is Onix. In Onix the Network Information Base (NIB) is used, in order to maintain the global network state.

The distributed architecture adopted by Onix offers a programmatic interface for the control logic from which operations via the connectivity infrastructure are conducted. The flat design is useful to avoid having a single point of failure in the SDN domain, but the overhead of the extra control and the complexity required for the controller management represent a disadvantage with respect to a hierarchical design.

A typical hierarchical controller structure is implemented by Kandoo. As shown in figure 12, in Kandoo, a two-layer controller is implemented, where a root controller is communicating with the domain controllers, avoiding the communication among controllers belonging to the same layer.

Figure 13 shows the architecture of OpenDaylight controller, which is one of the SDN controllers that better fit in a Hybrid ICN infrastructure thanks to its model-driven abstraction layer, as shown in vICN [52].

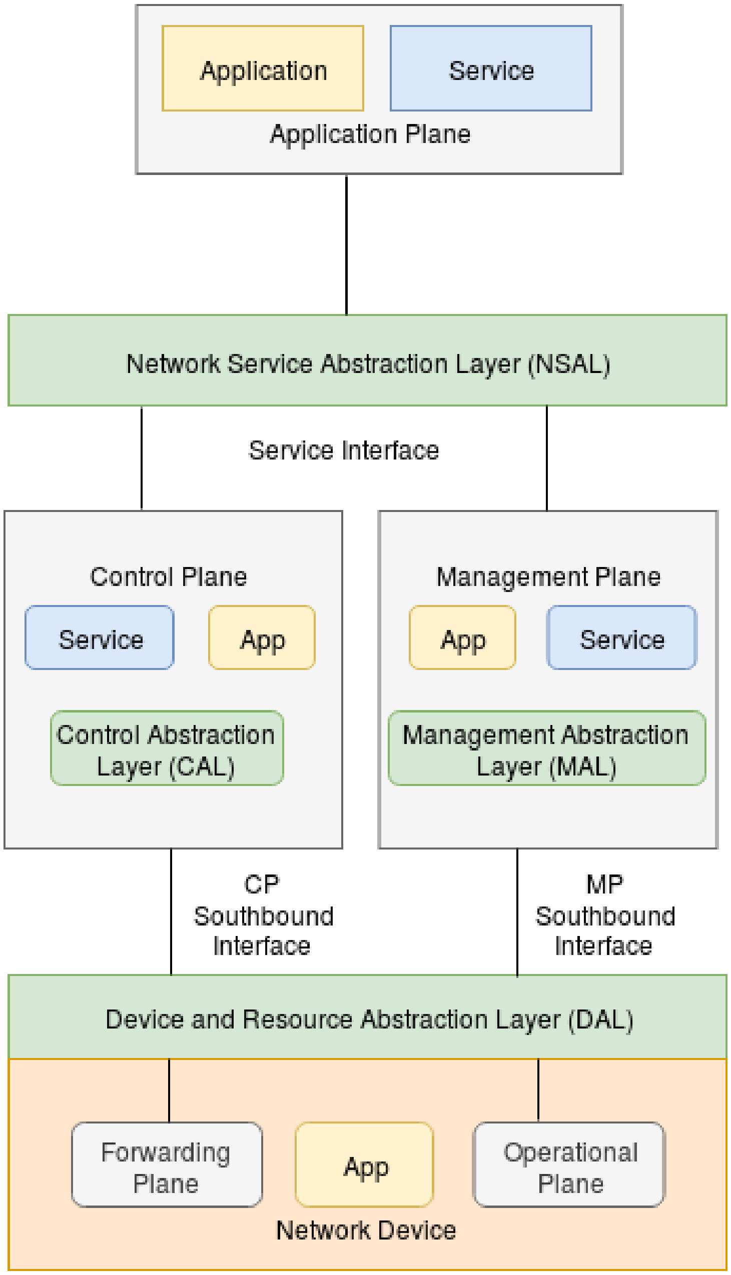

Figure 10: SDN Layer architecture

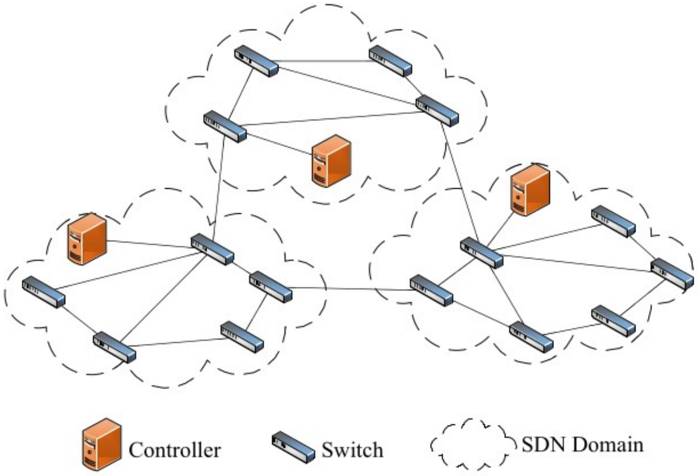

Figure 11: Multi-controller flat design [62]

In [63] Sardara et al. showcased a transport layer and socket API that follows the successful BSD socket approach and can be used in several ICN architectures, such as NDN, Content Centric Network (CCN) and hybrid Information Centric Networks (hICN).

5.2.4 Information Centric Networks

The issues raised by the increase of video traffic and the quantity of devices enabled to the communication, need to be faced by Internet Service Provider (ISPs) and Over The Top (OTTs) providers, but at the same time also the research community has been motivated to explore new designs able to provide a more scalable Internet, having as main goal an efficient content delivery.

One of the results of these efforts is the ICN paradigm [64]. Two important features introduced by ICN are the native multicast and the model of data integrity obtained by securing the information itself, which instead of relying only on the security host to host provided by the certificates in the traditional approach, exploits hashing mechanism to secure information.

Security in ICN is implemented directly on the information and not only on the host that is providing it. This is due to the fact that the information requested by the nodes can be stored elsewhere in the network. For this reason a mechanism to secure it has been provided. The security of the information is implemented through the use of hash, where the content is combined with the public key of the publisher by ensuring data integrity.

In the nodes security is implemented by using certificates [65]. This approach guarantees more security than the one adopted in the traditional IP networks, where certificates are just used server side, while hashing techniques can be optionally

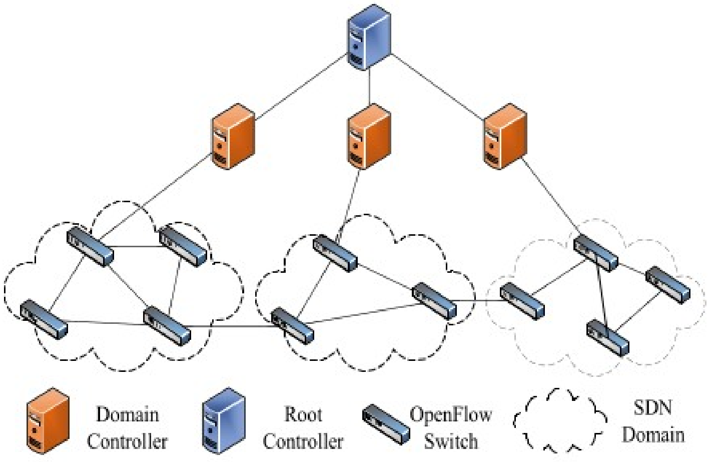

Figure 12: Hierarchical design of the multi-controller [62]

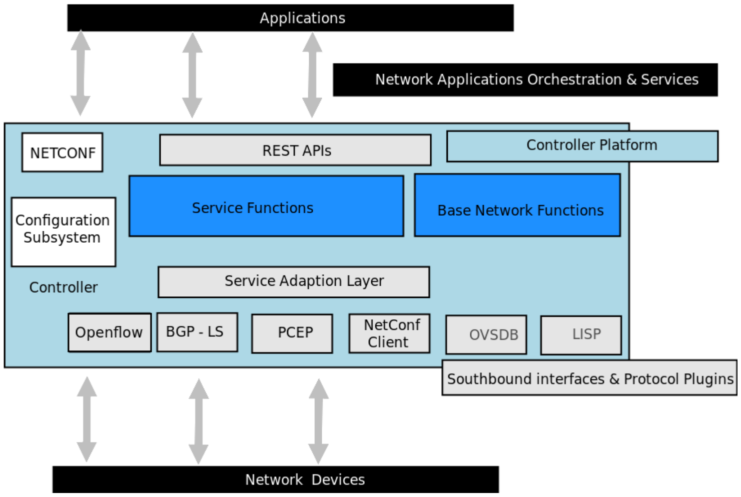

Figure 13: OpenDaylight architecture

used to ensure data integrity. In traditional IP network the certificates are mostly used to cipher the connection between host and server, but no mechanism to ensure data integrity is automatically adopted by the network itself.

On the other hand, multicast can be performed through the capabilities of the nodes to act as forwarders of information and through the in-cache functionalities provided by the network. In fact when one content is requested by more consumers, first it is sent to an area served by network in-cache functionality, then it will be sent to the consumer that requested it, but also will be cached for the next requests [66].

This permits high scalability and the offload of the original server avoids sending more times the same content to the same area in the same period. ICN offers a publish/subscribe paradigm which is implemented upon multicast. Namely, a multicast listener joins a multicast group to receive its data by subscribing to that specific group. Then, when a new content is published, the listeners will receive the requested data thanks to their previous subscription.

In traditional networks the adoption of multicast is limited just to intra-domain, and can not be easily used by users. In order to permit a host to receive multicast traffic, the router which is served by has to be configured to deliver the traffic related to a particular multicast group address, which is done using the Internet Group Management Protocol (IGMP), and the other routers in the path have to let that traffic cross.

Furthermore, the concept of multicast which is at the basis of ICN is more powerful than the traditional networks one, because it is based on the use of names instead of the IP addresses.

5.2.5 How blockchain can improve next generation architectures

RINA and SDN represent two different approaches where the abstraction of physical components plays an important role in order to face the problem of resource allocation. While the first is presented as a conceptual model for the implementation of architectures based on IPC, the second one represents a paradigm widely implemented by several open and closed source technologies.

No strict connnection among RINA and SDN has been made in literature, namely I did not find any works where SDN has been modeled by using the IPC architecture on which RINA is based. However both architectures can benefit of the blockchain technology in different ways.

Taking as example the Bitcoin blockchain [46], where a logical topology is implemented over a physical channel, it is important to mention how the transmission of the blockchain can be simplified using an ICN paradigm, such as, for example, the NDN implementation of ICN, with respect to the traditional IP network.

As shown in BlockNDN [37], implementing a blockchain ICN based brings several advantages, such as overhead reduction, minor convergence time and easier recovery of the missing blocks for the nodes that lost the synchronization with the blockchain.

Since NDN has no conception of IP addresses, the nodes just request data by sending interest packets, which are forwarded along the physical paths by the neighbors until it is discovered who has the request data. Whenever a node is able to send back the data requested, data packets will be forwarded to the reverse path.

Thanks to the in-cache functionalities and multicast transmissions implemented by publish/subscribe methods, the retrieval of the missing blocks, or of the whole blockchain in case of new nodes are joining the blockchain, is less energy consuming. As shown in DistBlockNet [53] the SDN controller is able to react actively to network threats such as DDoS/DoS attacks, cache poisoning/ARP spoofing and also able to detect security threats, by adopting their blockchain technique.

In their implementation they have shown how it is possible to use SDN and blockchain in order to prevent malicious behavior and to scale the network resources as needed. Sharma et al. in [54] focus their work on a blockchain where services are provided and billed by allowing transactions among consumers and providers. They defined their own consensus protocol, Proof of Service, which is the one used to forge blocks and chain them together. The architectural model proposed in RINA can be enriched by a similar concept.

In particular the distributed IPC facility that they proposed in [50], is able to provide not only fundamental services offered in traditional networking, but also services of application relaying, transaction processing and peer-to-peer. These last two elements are supporting elements of the blockchain technology.

The main proposal of the distributed IPC facility is to remove the barriers created by the present Transport Layer adopted in the current Internet. This removal can be performed by opening potential new markets for ISPs, enabling them to offer IPC services directly to costumers, increasing the value of their offer, thanks to the experience that they gain providing lower layers resource management.

The multi-control approach that is required by SDN to handle the increase of the network size analyzed by Hu et al. in their survey [62], can be extended by creating a communication layer blockchain based, where information regarding the network resources is exchanged in a safe way.

Applying a multi-control approach with blockchain capabilities to a fog computing scenario, new markets can be opened. Services can be offered by adopting cryptocurrencies to bill for the increase of the network resources provided and smart contracts can be used to regulate these interactions. SDN controllers can also operate remotely while keeping the provision of their functions fluently, if the link with the SDN network that they are controlling has enough capacity and is close to the consumer.

The adoption of the blockchain layer is able to create markets where consumers can buy SDN services via fog computing to those providers that demonstrate to have the expertise, improving performances in their local networks. In order to create the conditions for a better resource provisioning, it would be necessary that in the destination networks were deployed next generation networks able to exploit the enhanced functionalities offered by the latter.

In figure 14, I introduced the blockchain layer in the SDN Layer architecture presented in [58]. I encapsulated the Network Service Abstraction Layer (NSAL) in the introduced Blockchain Layer (BL). By encapsulating NSAL in BL it is possible to store in blocks each of its interactions with the lower layer performed by service interfaces, namely the Control and the Management planes, and at the same time to encapsulate the data exchanged with the Application plane in transactions contained

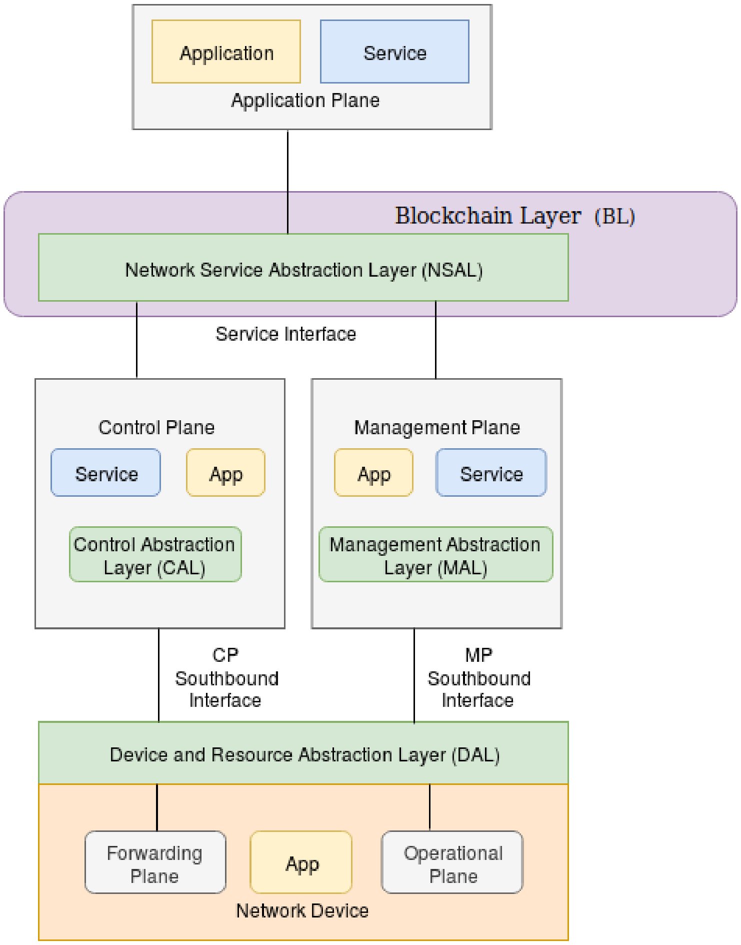

Figure 14: SDN Layer architecture with the Blockchain Layer

in blocks.

The interaction with the SDN controller, can be performed by invoking REST APIs that reside on top of its architecture, such as shown in Figure 4 in the OpenDaylight architecture. These APIs are also referred to as northbound APIs. In a hybrid approach where ICN is combined with SDN, a blockchain ICN oriented may even be adopted to manage the underlying SDN architecture, while keeping the improvements brought by its ICN implementation, such as less convergence time, less overhead and more granularity.

The adoption of a distributed ledger in a multi-control approach provides the consistency and the reliability required, inasmuch the ledger is by nature distributed among all the participants and not tempered. At the same time, by adopting a blockchain, it is possible to measure the network resources required and pay for them or propose an exchange with other unused resources, addressing scalability and load balancing issues.

The transactions contained in the blocks can store any kind of information that can be requested in order to improve the network management, such as requests for more network resources, as well as offers of unused network resources, routing tables to provide consistency, update on firewall policies to prevent attacks and so on. Smart contracts can be used in order to perform particular actions if a certain condition is verified. For example specific smart contracts for the management of the QoS can be defined.

When a certain value for a certain flow, or a group of flows, is under the desired threshold, the smart contract is executed in order to increase the network resources required.

In [2] I have defined REST APIs for the QoS, that can be implemented in the northbound of the SDN controller. By invoking methods such as GET /qos/availablebw?network=IPv4/netmask it is possible to retrieve values related, for example, to the available bandwidth devoted to a specific host, or a network. By invoking the method GET /qos/minmaxrtt?network=IPv4/netmask the minimum and maximum Round Trip Time (RTT) for a specific host or network can be retrieved.

This kind of approach can be helpful in the definition of smart contracts, in order to measure network requirements. In permissioned blockchain the identification of the entities involved in the communication can be performed by validators, or in some specific cases, such as in Hyperledger fabric, by certification authorities.

5.3 Future directions

I analyzed several works that implemented blockchain, giving a challenging point of view of its possible applications and integration with next generation architectures. I am focusing my research towards the investigation of ICN blockchain because I strongly believe that hybrid ICN architectures can be the solution of the resource allocation problem, paying particular attention to the QoS delivery for multimedia applications.

6 Conclusions

This thesis has faced the QoS related problems in real time communications, devoting particular attention to ISPs cooperative scenarios. At the beginning a deep analysis of the state of the art has been carried on, examining several methodologies available in literature. When it has been possible, open source paradigms have been evaluated in local or geographically distributed testbeds or by using their simulators.

In particular, networking paradigms such as SDN, NDN and CICN have been experimented. Furthermore, the analyzed subjects related to the QoS have been deepened and part of these results have been published in international congresses.

The activity of research and development presented in this thesis have permitted to elaborate, evaluate and understand the state of the art of the solutions offered to the QoS related problems in real time communications. For the blockchain deployment it has been hard to produce simulation results or to deploy the proposed design in a testbed.

Recently, an Amazon Web Service customized version of Hyperledger Fabric has been analyzed, and a possible implementation of the proposed ISP blockchain design could be performed, by exploiting the cloud infrastructure offered by AWS. Part of the source code, written with CloudFormation, for this deployment has already been tested successfully, and in the near future it will be extended in order to support my blockchain experimentation.

References

[1] M. Scarlato and C. Perra, “Next generation architecture for real time multimedia applications,” in *2017 AEIT International Annual Conference*. IEEE, 2017, pp. 1–6.

[2] M. Scarlato, J. Ortiz, C. Perra, and A. Skarmeta, “Leveraging OTT and ISP cooperation to enhance end to end QoS by exchanging valuable resources,” in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2018, pp. 118–120.

[3] M. Scarlato, C. Perra, and H. N. Lee, “A permissioned ISPs blockchain for the end to end Quality of Service enhancement,” in *Korean Institute of Electronics Engineers 2019 Summer Conference*, 2019.

[4] M. A. El-Gendy, A. Bose, and K. G. Shin, “Evolution of the Internet QoS and support for soft real-time applications,” *Proceedings of the IEEE*, vol. 91, no. 7, pp. 1086–1104, 2003.

[5] C. Perra and D. D. Giusto, “Light field compression on sliced lenslet array,” *International Journal of Internet Technology and Secured Transactions*, vol. 8, no. 1, pp. 48–61, 2018.

[6] C. Perra and D. Giusto, “Raw light field image compression of sliced lenslet array,” in *2017 IEEE international symposium on broadband multimedia systems and broadcasting (BMSB)*. IEEE, 2017, pp. 1–5.

[7] ——, “JPEG 2000 compression of unfocused light field images based on lenslet array slicing,” in *2017 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2017, pp. 27–28.

[8] C. Perra, F. Murgia, and D. Giusto, “An analysis of 3D point cloud reconstruction from light field images,” in *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, 2016, pp. 1–6.

[9] C. Perra, “A framework for user control over media data based on a trusted point,” in *2015 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2015, pp. 1–2.

[10] C. Westphal, “Challenges in networking to support augmented reality and virtual reality,” *IEEE ICNC*, 2017.

[11] C. Westphal, S. Lederer, D. Posch, C. Timmerer, A. Azgin, W. Liu, C. Mueller, A. Detti, D. Corujo, J. Wang, M. Montpetit, and N. Murray, “Adaptive video streaming over information-centric networking (icn),” Internet Requests for Comments, RFC Editor, RFC 7933, August 2016.

[12] A. Chanda, C. Westphal, and D. Raychaudhuri, “Content based traffic engineering in software defined information centric networks,” in *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2013, pp. 357–362.

[13] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, “Network coding: an instant primer,” *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 63–68, 2006.

[14] A. M. Sheikh, A. Fiandrotti, and E. Magli, “Distributed scheduling for low-delay and loss-resilient media streaming with network coding,” *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2294–2306, 2014.

[15] R. Mekuria, K. Blom, and P. Cesar, “Design, implementation, and evaluation of a point cloud codec for tele-immersive video,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 828–842, 2016.

[16] A. Al-Shuwaili and O. Simeone, “Energy-efficient resource allocation for mobile edge computing-based augmented reality applications,” *IEEE Wireless Communications Letters*, vol. 6, no. 3, pp. 398–401, 2017.

[17] R. Ravindran, X. Liu, A. Chakraborti, X. Zhang, and G. Wang, “Towards software defined ICN based edge-cloud services,” in *2013 IEEE 2nd International Conference on Cloud Networking (CloudNet)*. IEEE, 2013, pp. 227–235.

[18] T. Humernbrum, F. Glinka, and S. Gorlatch, “Using software-defined networking for real-time internet applications,” in *Proceedings of the International Multi-conference of Engineers and Computer Scientists*, vol. 1, 2014.

[19] J. Jo, S. Lee, and J. W. Kim, “Software-defined home networking devices for multi-home visual sharing,” *IEEE Transactions on Consumer Electronics*, vol. 60, no. 3, pp. 534–539, 2014.

[20] J. Suh, T. T. Kwon, P. Martinez-Julia, A. Skarmeta, T. You, L. Baron, S. Fdida, and J. Kim, “Enabling SDN experimentation with wired and wireless resources: The smartFIRE facility,” in *Cloud Computing: 6th International Conference, CloudComp 2015, Daejeon, South Korea, October 28-29, 2015, Revised Selected Papers*, vol. 167. Springer, 2016, p. 280.

[21] T. Rakotoarivelo, M. Ott, G. Jourjon, and I. Seskar, “OMF: a control and management framework for networking testbeds,” *ACM SIGOPS Operating Systems Review*, vol. 43, no. 4, pp. 54–59, 2010.

[22] Z. Zhu, C. Bian, A. Afanasyev, V. Jacobson, and L. Zhang, “Chronos: Serverless multi-user chat over ndn,” *Technical Report NDN-0008*, 2012.

[23] K. Choumas, T. Korakis, H. Lee, D. Kim, J. Suh, T. T. Kwon, P. Martinez-Julia, A. Skarmeta, T. You, L. Baron *et al.*, “Enabling SDN experimentation with wired and wireless resources: The smartfire facility,” in *International Conference on Cloud Computing*. Springer, 2015, pp. 280–290.

[24] “ndnSIM documentation.” [Online]. Available: https://ndnsim.net/current/

[25] M. Team, “Mininet.” [Online]. Available: http://mininet.org/

[26] V. GUEANT, “iperf - the ultimate speed test tool for tcp, udp and sctptest the limits of your network internet neutrality test.” [Online]. Available: https://iperf.fr/

[27] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, “Optimizing 360 video delivery over cellular networks,” in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*. ACM, 2016, pp. 1–6.

[28] J. Seppänen, M. Varela, and A. Sgora, “An autonomous QoE-driven network management framework,” *Journal of Visual Communication and Image Representation*, vol. 25, no. 3, pp. 565–577, 2014.

[29] E. Liotou, G. Tseliou, K. Samdanis, D. Tsolkas, F. Adelantado, and C. Verikoukis, “An SDN QoE-service for dynamically enhancing the performance of OTT applications,” in *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*. IEEE, 2015, pp. 1–2.

[30] F. Ongaro, E. Cerqueira, L. Foschini, A. Corradi, and M. Gerla, “Enhancing the quality level support for real-time multimedia applications in software-defined networks,” in *2015 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2015, pp. 505–509.

[31] R. Durner, A. Blenk, and W. Kellerer, “Performance study of dynamic QoS management for OpenFlow-enabled SDN switches,” in *2015 IEEE 23rd International Symposium on Quality of Service (IWQoS)*. IEEE, 2015, pp. 177–182.

[32] A. Ahmad, A. Floris, and L. Atzori, "QoE-centric service delivery: A collaborative approach among OTTs and ISPs," *Computer Networks*, vol. 110, pp. 168–179, 2016.

[33] C. Cachin, "Architecture of the hyperledger blockchain fabric," in *Workshop on distributed cryptocurrencies and consensus ledgers*, vol. 310, 2016.

[34] S. Gec, D. Lavbič, M. Bajec, and V. Stankovski, "Smart contracts for container based video conferencing services: Architecture and implementation," *arXiv preprint arXiv:1808.03832*, 2018.

[35] U. Paščinski, J. Trnkoczy, V. Stankovski, M. Cigale, and S. Gec, "QoS-aware orchestration of network intensive software utilities within software defined data centres," *Journal of Grid Computing*, vol. 16, no. 1, pp. 85–112, 2018.

[36] G. Sedky and A. El Mougy, "BCXP: Blockchain-centric network layer for efficient transaction and block exchange over Named Data Networking," in *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*. IEEE, 2018, pp. 449–452.

[37] T. Jin, X. Zhang, Y. Liu, and K. Lei, "BlockNDN: A bitcoin blockchain decentralized system over named data networking," in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2017, pp. 75–80.

[38] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[39] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *IEEE P2P 2013 Proceedings*. IEEE, 2013, pp. 1–10.

[40] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-ng: A scalable blockchain protocol," in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI) 16*, 2016, pp. 45–59.

[41] Z. Zhu and A. Afanasyev, "Let's chronosync: Decentralized dataset state synchronization in named data networking," in *2013 21st IEEE International Conference on Network Protocols (ICNP)*. IEEE, 2013, pp. 1–10.

[42] J. Chen, M. Arumaithurai, L. Jiao, X. Fu, and K. Ramakrishnan, "Copss: An efficient content oriented publish/subscribe system," in *Proceedings of the 2011 ACM/IEEE Seventh Symposium on Architectures for Networking and Communications Systems*. IEEE Computer Society, 2011, pp. 99–110.

[43] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail," in *Annual International Cryptology Conference*. Springer, 1992, pp. 139–147.

[44] B. M. Jakobsson and A. Juels, "Proofs of work and bread pudding protocols," Apr. 8 2008, uS Patent 7,356,696.

[45] A. Back *et al.*, "Hashcash-a denial of service counter-measure," 2002.

[46] S. Nakamoto *et al.*, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[47] K. Iyer and C. Dannen, *Building Games with Ethereum Smart Contracts*. Springer, 2018.

[48] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," 2016.

[49] V. Buterin and V. Griffith, "Casper the friendly finality gadget," *arXiv preprint arXiv:1710.09437*, 2017.

[50] J. Day, I. Matta, and K. Mattar, "Networking is IPC: a guiding principle to a better internet," in *Proceedings of the 2008 ACM CoNEXT Conference*. ACM, 2008, p. 67.

[51] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl, "Network of information (netinf)–an information-centric networking architecture," *Computer Communications*, vol. 36, no. 7, pp. 721–735, 2013.

[52] M. Sardara, L. Muscariello, J. Augé, M. Enguehard, A. Compagno, and G. Carofiglio, "Virtualized ICN (vICN): towards a unified network virtualization framework for ICN experimentation," in *Proceedings of the 4th ACM Conference on Information-Centric Networking*. ACM, 2017, pp. 109–115.

[53] P. K. Sharma, S. Singh, Y.-S. Jeong, and J. H. Park, "Distblocknet: A distributed blockchains-based secure SDN architecture for IoT networks," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 78–85, 2017.

[54] P. K. Sharma, M.-Y. Chen, and J. H. Park, "A software defined fog node based distributed blockchain cloud architecture for IoT," *IEEE Access*, vol. 6, pp. 115–124, 2018.

[55] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[56] N. M. K. Chowdhury and R. Boutaba, "Network virtualization: state of the art and research challenges," *IEEE Communications magazine*, vol. 47, no. 7, pp. 20–26, 2009.

[57] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang, "A clean slate 4D approach to network control and management," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 5, pp. 41–54, 2005.

[58] E. Haleplidis, K. Pentikousis, S. Denazis, J. Salim, D. Meyer, and O. Koufopavlou, "RFC 7426: Software-Defined Networking (SDN): layers and architecture terminology," *Internet Research Task Force (IRTF)*, 2015.

[59] A. Dixit, F. Hao, S. Mukherjee, T. Lakshman, and R. Kompella, "Towards an elastic distributed SDN controller," *ACM SIGCOMM computer communication review*, vol. 43, no. 4, pp. 7–12, 2013.

[60] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow *et al.*, "ONOS: towards an open, distributed SDN OS," in *Proceedings of the third workshop on Hot topics in software defined networking.* ACM, 2014, pp. 1–6.

[61] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama *et al.*, "Onix: A distributed control platform for large-scale production networks." in *OSDI*, vol. 10, 2010, pp. 1–6.

[62] T. Hu, Z. Guo, P. Yi, T. Baker, and J. Lan, "Multi-controller based software-defined networking: a survey," *IEEE Access*, vol. 6, pp. 15 980–15 996, 2018.

[63] M. Sardara, L. Muscariello, and A. Compagno, "A transport layer and socket api for (h) ICN: Design, implementation and performance analysis," in *Proceedings of the 5th ACM Conference on Information-Centric Networking (ACM ICN'18)*, 2018.

[64] R. Tourani, S. Misra, T. Mick, and G. Panwar, "Security, privacy, and access control in information-centric networking: A survey," *IEEE communications surveys & tutorials*, vol. 20, no. 1, pp. 566–600, 2018.

[65] D. Kutscher *et al.*, "RFC 7927: ICN research challenges," *IRTF, ICNRG*, 2016.

[66] K. Su, F. Bronzino, K. Ramakrishnan, and D. Raychaudhuri, "MFTP: A clean-slate transport protocol for the information centric mobilityfirst network," in *Proceedings of the 2nd ACM Conference on Information-Centric Networking.* ACM, 2015, pp. 127–136.