Peer-to-Peer Blockchain-based NFV Service Platform for End-to-End Network Slice Orchestration Across Multiple NFVI Domains

Pol Alemany, Ricard Vilalta, Raul Muñoz, Ramon Casellas, Ricardo Martínez
Optical Network & Systems Dept.
Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA)
Castelldefels, Spain
palemany@cttc.cat

***Abstract*—This paper presents a non-hierarchical architecture to deploy End-to-End Network Slices in a multi-domain network using an Ethereum-based Blockchain to manage the Network Slicing requests across domains. The use of Blockchain aims to look towards a collaboration vision to deploy Networks Slices using the resources to deploy them as if they would be placed under the domain of the Network Slice requester. The authors describe a possible instantiation procedure and they present results showing how much the use of Blockchain might increase the deployment time of an End-to-End Network Slice.**

***Index Terms*—Network Slice, Blockchain, NFV**

I. Introduction

Nowadays multiple telecommunications network architecture designs co-exist in order to transmit information from one point to another through deployed services. The latest network architectures focus on implementing Software-Defined Networks (SDN) and Network Function Virtualisation (NFV) standards as they allow a more flexible and dynamic configuration of the networking and computing resources. The use of SDN allows the differentiation between data and control planes. The most common model used when designing control planes is the hierarchical model as presented in [1], with a component on the top able to manage and control the whole network under its domain. Another model is the mesh model in which all nodes are equal and they work in a collaborative way, creating what is called a peer-to-peer (P2P) network. P2P networks have been used in multiple applications such as files exchange systems [2] [3], researched to be used for Voice on Demand services [4] among other possibilities.

One of the main problems of P2P networks was the trust between peers. While in centralised architectures like the cloud model trust may be given and evaluated as presented in [5] due to the existence of a central authority, in P2P networks this is not possible. So, any received data could be corrupted so that once executed or opened would harm our devices. Some years ago Blockchain started allowing the use of P2P networks for different services from financial exchanges with Bitcoin [6] to new applications using Ethereum [7].

Work partially funded by the EC through the 5GPPP INSPIRE-5GPlus (871808) and MINECO AURORAS (RTI 2018-099178-B-I00) projects.

Blockchain is a Distributed Ledger Technology (DLT): a digital system that records asset transactions -i.e. money, resources, information- by saving the transactions and their detail in different places at the same moment. It might be understood as a distributed database (DB) with all nodes -i.e. peers- keeping the same information. Blockchain allows to update the information in an iterative and secure way. When a transaction is done, its information and related metadata are saved in all nodes, making them all aware of that information and making it impossible to modify it without the others nodes knowing it. The main characteristics of Blockchain are:

- Distributed: As the data is distributed and there is no central authority, the system is robust against hacks.
- Secure: All information in the DB is encrypted using private and public keys.
- Public: The system is more transparent as there is no central authority to track and validate all the information, but all peers do it.

Blockchain has been used already to demonstrate possible applications to manage SDN/NFV networks, the idea of using Blockchain in a multi-domain environment and three possible scenarios is presented in [8]. In [9], Blockchain allows to share the information among a set of optical switches to calculate the best path possible across them. Furthermore, [10] makes use of Blockchain to keep track of Service Level Agreements (SLA) events over a disaggregated network and finally, [11] describes an algorithm using Blockchain to quickly configure switches to be controlled by the most optimal master when their initial master goes down or becomes evil. In the previous papers and most of the networks and Blockchain literature, the focus is on the management of physical resources -e.g optical path calculation, traffic SLA fulfillment and switches management-, but there is few research looking into higher layer elements such as Network Services (NSs) or Network Slices (Slices).

Network Slicing as described in [12] is a backbone for the future networks management. Using the definition made by the 3GPP [13], the authors of this paper have presented the benefits of Network Slicing on the SDN/NFV networks management in previous works. From a very basic scenario

defining a Network Slice Manager [14] (referred to as Slicer), to one of the latest evolutions presented in [15] where a Slice is composed by Virtual and Cloud-native Network Functions (VNFs and CNFs respectively). All these previous works had in common that there was just a single controller to manage the virtual resources in a multi-domain scenario following a hierarchical architecture. If the domains belong to different entities, the use of a hierarchical architecture might create disputes as only one single entity on the top has the control of the network. Using Blockchain may be a possible solution in order for each entity to keep the control of its own domains but, at the same time, to have a fair and equal collaboration among domain owners to deploy services.

As previously said, there is few research joining Blockchain and Network Slicing. Some examples are [16] and [17] which focus on the use of of Blockchain to create End-to-End (E2E) Slices in a hierarchical architecture with a single Slicer on the top over the different network domains. This paper presents a multi-domain NFV/SDN network, in which each domain has its own NFV/SDN architecture with a Slicer on the top. So the Slicers in each domain, collaborate among them using Blockchain to deploy E2E Slices across domains. The main idea is that each Slicer shares its own Network Slice Templates (NSTs) -which may be a single Slice or a part of a Slice (slice-subnet)- and network resources in its domain with the other domains. So, if a vertical in a domain needs a service controlled by a different domain Slicer, it must simply request the E2E Slice to its domain Slicer which takes care of the whole Slice deployment with the collaboration of the other Slicers using Blockchain.

This paper is organized as follows: section II introduces the idea of using P2P networks to manage Network Slicing with the use of Blockchain; Section III details the steps procedure to deploy an E2E Slice in a Network Slicing multi-domain architecture; Section IV describes the experimental evaluation in an emulated environment of the architecture presented and, finally, section V presents the final conclusions.

II. Designing a Collaborative Network Slicing Infrastructure for a Multi-Domain Network

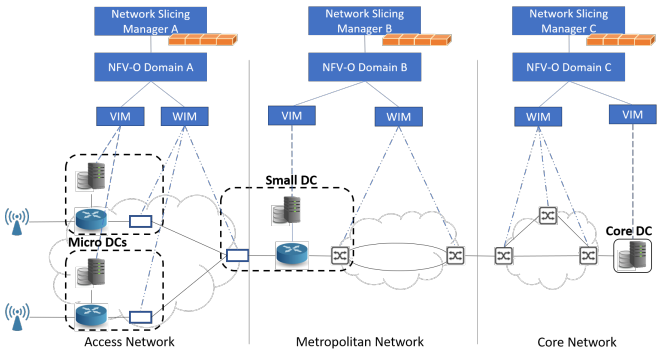

Fig. 1. Blockchain P2P Architecture.

This section presents the multi-domain architecture designed to have a collaborative network that allows the E2E Slices deployment across domains. In order to do so, the Slicers are members of a private Blockchain in which they can share their own resources with the other domains in a reliable way.

Figure 1 presents the architecture used to create a P2P network in which each peer shares its available Network Slicing and computing resources to instantiate part of an E2E Slice requested by another peer. All domains follow the same architecture presented in [18] in which the ETSI merged the 3GPP Network Slicing proposal with its standardised NFV architecture.

On the top of each orchestration domain there is a Slicer in charge of all the Network Slicing related actions while being a peer of the Blockchain. Each Slicer is the owner of the resources in its domain, but the management of these resources changes depending on whether the instantiation is requested by the Slicer in the same domain or by a Slicer in a different domain. If a Vertical requests an E2E Slice instantiation to its domains Slicer and it only uses resources placed within the same Slicer domain, that Slicer is the unique owner and, through the NFV Orchestrator (NFVO), it can apply any necessary E2E Slice related action to the resources in the network. On the other hand, if a vertical requests an E2E Slice instantiation using resources of different domains, the Slicer which has requested the E2E Slice is the unique owner, but any action to apply to the resources placed in other domains must go through the Blockchain and then, to the other Slicers. These will then request their NFVOs to apply the corresponding actions to their domain network resources. Keeping this in mind, if a Slicer has some of its domain network resources being used for an E2E Slice requested by another domain Slicer, these computing resources cannot be modified unless the E2E Slice owner asks for it.

As previously presented, the Slicer component has a secondary but essential functionality for the collaborative architecture to work: being a peer in the Blockchain network. Despite Blockchain being known as a secure and trustworthy technology and one of its main key stones is to be a public database, it is also possible to have a Blockchain with only a certain set of peers allowed to be part of it. This last idea is how the proposed architectures makes use of Blockchain: only the Slicers belonging to known domains can be a peer in the collaborative network. Despite the Blockchain being private, the rest of its characteristics are kept. So, there is no central point of authority -i.e. no Slicer has more power than the others-, the information in the Blockchain is public -i.e. if a peer replies a NST from another peer, all the peers will realise-, and all information is still secure as it is encrypted and can be read only by the accepted peers.

Under each Slicer (and also Blockchain) component, in the middle and lowest levels of the architecture presented in figure 1 there are: first, the NFVOs in charge of the NSs and VNFs lifecycle management and orchestration. Second, the Virtual and WAN Infrastructure Managers (VIMs and WIMs respectively) which are the SDN Controllers for the computing -i.e. VIMs- and networking -i.e. WIMS- resources

in the physical network.

III. Instantiation Procedure Description

While in a scenario in which there is a single Slicer all the Slice instantiation process is done in that single manager, now the process must involve all the interconnected managers through the use of the Blockchain they are part of.

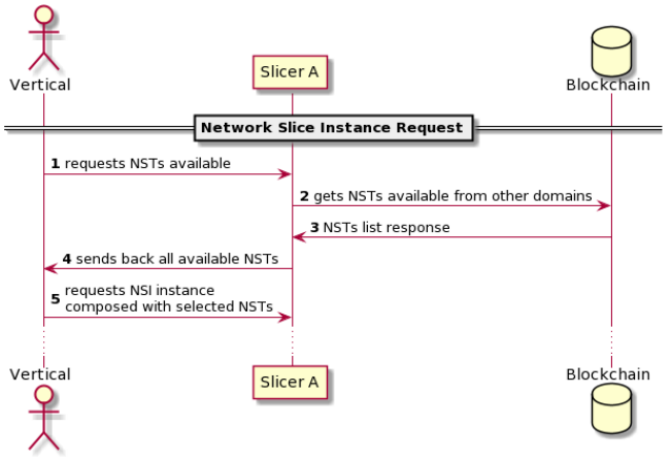

Fig. 2. NST catalogue and Network Slice instance requests step

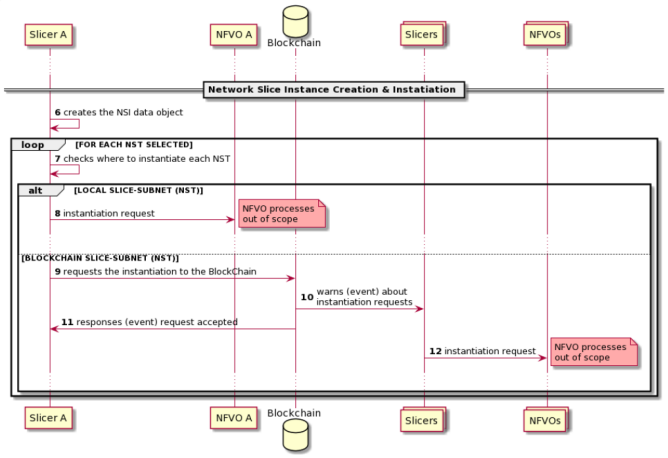

Fig. 3. Network Slice instance creation and deployment step

The complete process to deploy an E2E Slice in a multi-domain architecture involving different Slicers is divided in three parts: the first part describes the design of the E2E Slice, the second part presents the steps to instantiate the E2E Slice components and, finally, the third part involves the verification of a correct E2E Slice deployment. Figures 2, 3 and 4 show the flow diagram for each part respectively.

The deployment begins with the E2E Slice design steps in figure 2. When a 5G Vertical needs a Slice with a set of services requirements to be fulfilled, it requests (1) the catalogue with the available NSTs to the Slicer in its domain (Slicer A in the figure). Then, Slicer A will get its own NSTs from the local DB but it also requests the Blockchain the NSTs offered by the other Slicers (2). Once the Blockchain has passed the information (3), the Slicer A sends back to the Vertical all the possible options (4). Finally, the Vertical is in a position to request the E2E Slice composed by the selected NSTs (5).

The second part, which corresponds to the E2E Slice components deployment, begins with the Slicer A creating the E2E Network Slice Instance (NSI) object using the selected NSTs to compose the the internal elements (referred to as slice-subnets) of E2E Slice instantiation object -i.e. NSI- (6). Having created the NSI, the Slicer A checks all the slice-subnets information in the NSI to know whether the referenced NST is local or from another Slicer Domain (7). If the NST is local (8), the Slicer A requests the instantiation to its local NFVO -i.e. NFVO A- and this one does the required actions to create the corresponding virtual elements (out of the scope of this article). On the other hand, if the NST is external (9), the request is sent to the Blockchain and once it reaches the Blockchain, two actions are triggered: first the Blockchain warns the NST owner (Slicer B) with an event to instantiate the selected NST (10), and then the Blockchain answers back to Slicer A to inform that the process is going on (11). Meanwhile, the other Slicers request their NFVOs the NST instantiation (12) and the NFVOs do the required actions to create the corresponding virtual elements (out of the scope of this article).

The third and last part corresponds to the verification steps to ensure that all the slice-subnets -i.e. virtual elements- composing the E2E Slice are deployed. This part begins with the Slicer A continuously checking the NSI to validate if all slice-subnets are instantiated (13) while, in parallel, two possibilities might occur: (14) it receives an update from its local NFVO (NFVO A) and updates the corresponding NSI slice-subnet information (15) or, on the other side, if the slice-subnet in any of the other NFVOs is ready, the corresponding Slicer is informed (16) and the the Slicer updates its local DB and the Blockchain (17). At this point, the Blockchain behaves similarly like in steps (10,11): first warns the Slicer A about the updated slice-subnet (18) and, second, it responses back (19) to the corresponding Slicer about the request in step 17 being processed. Then, the Slicer A updates the NSI with the incoming Blockchain information (20). Finally, like in step 13, the Slicer A validates if all slice-subnets are ready (21) if they are not, Slicer A keeps waiting for new updates to start again (13), otherwise the whole instantiation process is finished and the Vertical informed (22).

IV. Experimental Evaluation

This section introduces the reasons to select the chosen Blockchain technology. Then it shows the use case designed to test the collaborative network and, finally, it presents the results to demonstrate the instantiation procedure described in section III and the delay added by the Blockchain in that same procedure respect to previous works that also focused on the deployment of E2E Slices.

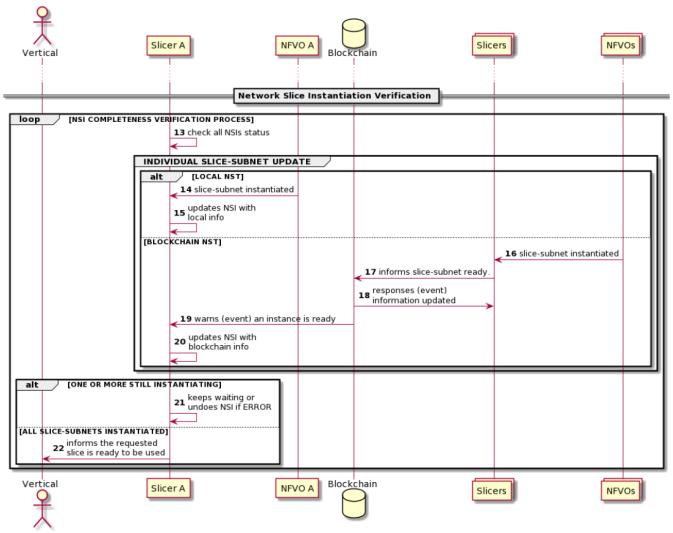

Fig. 4. Network Slice instance verification step

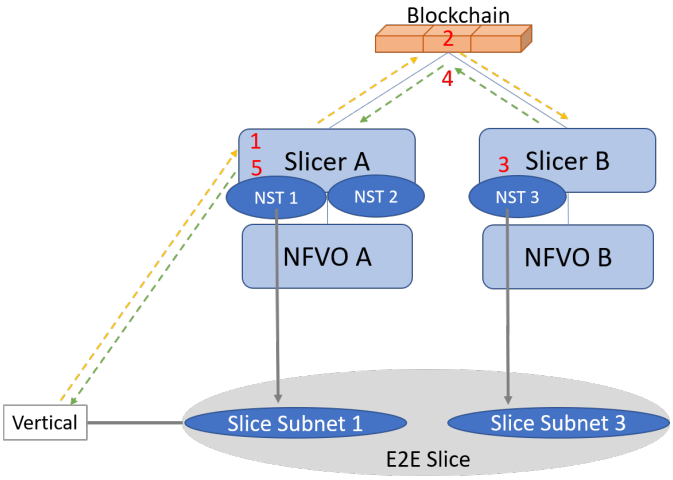

Fig. 5. Instantiation use case.

A. Blockchain selection

Among all the Blockchain possibilities the two most known are Bitcoin and Ethereum, and the second was the selected platform to be used. The reason is that aside of its cryptocurrency -i.e. Ether (ETH)- capabilities, it allows to design and create decentralised applications (dapps) using smart contracts -i.e. a set of functions to be done only inside the Blockchain- which are known by all the Blockchain nodes. In addition, Ethereum was designed with its own programming language called Solidity.

Another reason to select Ethereum instead of Bitcoin is the time required to create the data blocks. In fact, while Bitcoin uses SHA-256 and takes minutes to encrypt a block, Ethereum uses ethash and requires seconds, which would surely affect the results presented in this paper.

B. Use Case Description

The use case developed to demonstrate the flow described in section III is presented in figure 5 and it has the following architecture:

- A network with two domains and each domain with its own Slicer (Slicer A and Slicer B) and NFV infrastructure (NFVO A and NFVO B).
- Slicer A has two NSTs (NST_1 and NST_2) available.
- Slicer B has one (NST_3) available.
- A Blockchain with the two Slicers being its nodes -i.e. in Blockchain terms, the miners- and it contains resumed information of the NSTs available in each Slicer.

.

The objective of the use case is to evaluate if using a collaborative network scenario to deploy E2E Slices using Blockchain adds a significant delay respect to an E2E Slice deployment done in a hierarchical scenario with a single Slicer. The use case follows the flow diagram presented in section III and in order to evaluate the added time by the Blockchain, five time samples (red numbers in figure 5) were taken:

1) Slice-Deployment-T1: It is the time between the vertical request -i.e. E2E Slice composed with NST_1 and NST_3- reaches Slicer A and all the slice-subnets instantiation requests are requested either to the local NFVO -i.e. NST_1- A and to the Blockchain -i.e. NST_3-.
2) Blockchain-T2: The time for the Blockchain to process the request, to warn (for each slice-subnet) the involved Slicer to deploy the associated NST -i.e. Slicer B with NST_3- and to answer back to Slicer A to inform that the request is being processed.
3) Slice-Deployment-T3: It corresponds to the necessary time by Slicer B to process its instantiations and, once they are done, to update the information in the Blockchain -i.e. NST_3 is ready-.
4) Blockchain-T4: The time required to manage the updated information, to warn the Slicer A about the instantiations status managed by other Slicers -i.e. instance of NST_3 ready- and to answer back to Slicer B.
5) Slice-Deployment-T5: The time to process the last actions and leave the E2E Slice instance ready to be used by the Vertical.

The whole experimental process was done in an emulated scenario in which the Slicers were not sending the requests down to the NFVOs with the objective to obtain clearly show the influence of the time values in which the Blockchain layer was involved. The Blockchain was emulated using Ganache [19]. This software allows to create private Ethereum Blockchain in a controlled environment and in few steps while keeping the possibility to perform any possible action, as it would be done in a real Ethereum Blockchain, without any cost.

C. Results

Figures 6 and 7 present the HTTP traffic and the Ethereum transactions respectively, which demonstrate the instantiation procedure described in section III.

1	-------	Vertical A	Slicer A	HTTP	287	POST /add_nst HTTP/1.1 (application/json)
	-------	Vertical B	Slicer B	HTTP	292	POST /add_nst HTTP/1.1 (application/json)
	Requests to upload the NSTs to the Blockchain (Transactions A and B)					
2	-------	Vertical A	Slicer A	HTTP	157	GET /get_all_nst HTTP/1.1
3	*REF*	Vertical A	Slicer A	HTTP	352	POST /instantiate_nsi HTTP/1.1 (application/json)
4	0.074116446	Slicer A	NFVO A	HTTP	248	POST /instantiate_slice_subnet HTTP/1.1 (application/json)
	Slice-subnet instantiation request through Blockchain (Transaction C)					
5	5.932282587	Slicer B	NFVO B	HTTP	336	POST /instantiate_blockchain_slice_subnet HTTP/1.1 (application/json)
6	6.096653710	NFVO A	Slicer A	HTTP	248	POST /update_local_slice_subnet HTTP/1.1 (application/json)
	Slice-subnet updated coming from Blockchain (Transaction D)					

Fig. 6. HTTP traffic

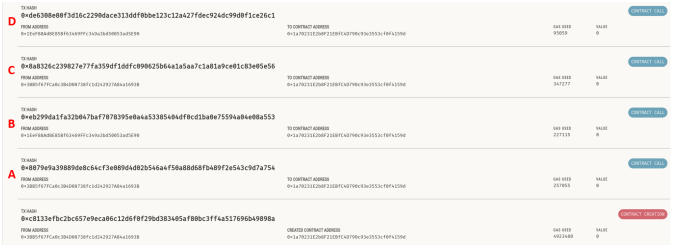

Fig. 7. Ethereum Transactions

First all NSTs must be added into the local DB (Fig. 6 step 1) of each Slicer and uploaded in the Blockchain (Fig. 7 transactions A and B). Then, 5G Verticals have all the NSTs available (Fig. 6 step 2). When one of them (Vertical A) requests the deployment of an E2E Slice (Fig. 6 step 3) to its Slicer (Slicer A), this creates the NSI object with its slice-subnets -i.e. selected NSTs- and requests their deployment to corresponding Slicer domain: its own NSTs are requested (Fig. 6 step 4) to its domain NFVO (NFVO A), while the external NSTs requests are sent to the corresponding Slicer -i.e. Slicer B- through the Blockchain (Fig. 7 transaction C). When Slicer B receives it, it creates an NSI to keep the local track of the computing resources used and requests (Fig. 6 step 5) to its local NFVO (NFVO B) the deployment of the NST components. Once all the subnet-slices of the E2E Slice are deployed, the E2E Slice owner -i.e. Slicer A- is informed directly by its local NFVO A (Fig. 6 step 6), or through the Blockchain (Fig. 7 transaction D) about those deployments done in other domains.

TABLE I
TIME STEPS STANDARD DEVIATION

σ (s)					
Total	Step 1	Step 2	Step 3	Step 4	Step 5
2.895452	0.162317	2.865800	0.092082	1.204695	0.000301

In order to evaluate how Blockchain may affect the required time to instantiate a Network Slice, the described use case deployment was tested and the results are presented in Fig. 8. This figure shows the mean values of each one of the five time samples defined in subsection IV-B and Fig. 5. In addition, Tab. I presents the corresponding standard deviation values of each one of the time samples.

As previously described, the tests were done in a environment in which the creation of the the virtual nodes and links were not done, this is why the mean value of the total instantiation time is of 9.627458s and columns 1/3/5 have such small values as these three columns are actions done locally in each NFV domain. By doing so, it is possible to see the increment of time added when a non-local slice-subnet is initially requested and the Blockchain must create its internal data object to keep track (column 2) and later, when this same data object needs to be updated (column 4).

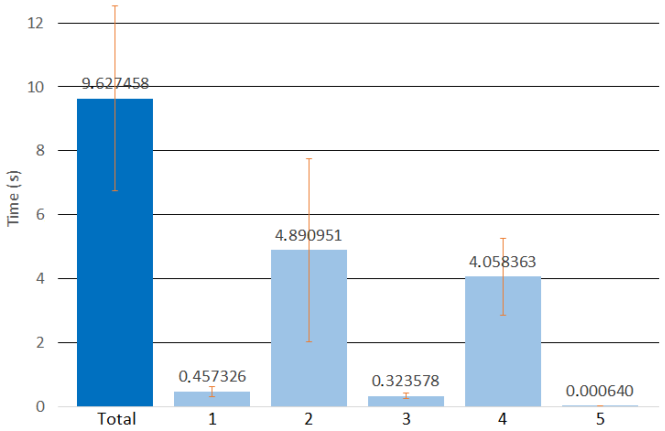

Fig. 8. Set Up Phases Delay Time

With the values presented in a previous work [15] which compared the influence of kernel-based Virtual Machines (kVM) and Containers creation on the instantiation of an E2E Slice. The mean value of the deployment time for kVM-based slice-subnets had a magnitude of 11 minutes, while for a set of container-based slice-subnets, it was around 87.5 seconds. Now, taking the worst case possible -i.e. total time plus the standard deviation- in this paper use case, the time value is of 12.52291 seconds. Taking into account that steps 1, 3 and 5 are less than a second and they are the steps when the slice-subnets are instantiated on the physical network, the steps in which the Blockchain influences add around 11.5 seconds. So, on a kVM-based slice-subnet instantiation, Blockchain is barely

noticed as 11 are 660 seconds, adding 11.5 seconds more the increment is of a 1.71%. But, in a container-based slice-subnet instantiation Blockchain, the percentage increment is of an 11.61%.

V. Conclusions

It is very complicated that different domain owners, with different objectives or requirements, might share their resources and trust the other domain owners to act equally and collaborate without any contract or any agreement between them. This paper focused on the use of an Ethereum Blockchain as a tool to demonstrate that it is possible to have a network architecture without the necessity of having a central authority managing the end-to-end network actions or behaving as a moderator in a multi-domain network with multiple points of view.

The architecture presented focuses on the management and orchestration of E2E Network Slices across different domains. The Network Slicing Manager is placed as the component on the top of the NFV/SDN infrastructure and, at the same time, it is the component that, through the Ethereum Blockhain, interacts with the other network domains. In addition to the architecture, a traffic analysis is provided in order to describe how the Network Slice Managers interact with each other. Furthermore, results are presented to show how the use of Blockchain may affect the deployment of E2E Network Slices across different domains.

The results probed that Blockchain might be a good option in a multi-domain scenario but, depending on which virtualisation technology -i.e. Virtual Machines or Containers- is used in the computing resource nodes, the increment of time added by the Blockchain operations might have a bigger influence to the total amount of time to deploy E2E Network Slices. The first aspect to take into account with the idea of having multiple Slicers collaborating is the avoidance of a central point of failure (CPoF). If in a multi-domains scenario, there is a single Slicer and it is attacked and taken down, then the management of E2E Slices would be not available. By using a multiple Slicers working together, this possibility is removed because the E2E Slice may be requested to another domain Slicer. A second aspect to consider when using this architecture is that the added time by the Blockchain compared to the avoidance of the CPoF might be a small sacrifice to do in order to gain some security. Moreover, it should be tested if using a centralised architecture does not add a similar time value to the whole deployment process.

While the current work shows that the use of Blockchain might help on multi-domain networks, this paper used a private Blockchain, which means that all the nodes are known among them. A future work based on this paper is the use of self-designed Ethereum Blockchain and the selection of the appropriate consensus mechanism for the network resources management and E2E Slices deployments. Another possibility as a future work is the use of a public Blockchain in order to gain flexibility -i.e. adding new domains without the need to known the owners-, this might not be possible because as in any business within the telecommunications field, trusting an entity that it is not known without any guaranty of a proper behaviour is a difficult task.

References

[1] R. Muñoz, R. Vilalta, R. Casellas, R. Martínez, F. Vicens, J. Martrat, V. López, and D. López, "Hierarchical and recursive NFV service platform for end-to-end network service orchestration across multiple NFVI domains," in *in Proc. of International Conference on Transparent Optical Networks (ICTON), 1-5 July 2018, Bucharest (Romania)*, Jul. 2018.

[2] Bittorrent. [Online]. Available: http://www.bittorent.com

[3] Emule. [Online]. Available: http://www.emule.com

[4] M. Fouda, T. Taleb, M. Guizani, Y. Nemoto, and N. Kato, "On supporting p2p-based vod services over mesh overlay networks," in *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, 2009, pp. 1–6.

[5] F. Di Cerbo, P. Bisson, A. Hartman, S. Keller, P. H. Meland, M. Moffie, N. G. Mohammadi, S. Paulus, and S. Short, "Towards trustworthiness assurance in the cloud," in *Cyber Security and Privacy Forum*. Springer, 2013, pp. 3–15.

[6] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Cryptography Mailing list at https://metzdowd.com*, 03 2009.

[7] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[8] R. V. Rosa and C. E. Rothenberg, "Blockchain-based decentralized applications for multiple administrative domain networking," *IEEE Communications Standards Magazine*, vol. 2, no. 3, pp. 29–37, 2018.

[9] H. Yang, Y. Li, S. Guo, J. Ding, Y. Lee, and J. Zhang, "Distributed blockchain-based trusted control with multi-controller collaboration for software defined data center optical networks in 5g and beyond," in *2019 Optical Fiber Communications Conference and Exhibition (OFC)*, 2019, pp. 1–3.

[10] S. Fichera, A. Sgambelluri, A. Giorgetti, F. Cugini, and F. Paolucci, "Blockchain-anchored failure responsibility management in disaggregated optical networks," in *2020 Optical Fiber Communications Conference and Exhibition (OFC)*, 2020, pp. 1–3.

[11] Y. Liang, H. Yang, Q. Yao, S. Guo, A. Yu, and J. Zhang, "Blockchain-based efficient recovery for secure distributed control in software defined optical networks," in *2019 Optical Fiber Communications Conference and Exhibition (OFC)*, 2019, pp. 1–3.

[12] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5g with sdn/nfv: Concepts, architectures, and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80–87, 2017.

[13] 3GPP, "Tr 28.801 (v. 15.1.0): Study on management and orchestration of network slicing for next generation network," 3GPP, Tech. Rep., 1 2018.

[14] R. Vilalta, P. Alemany, R. Casellas, R. Martínez, C. Parada, J. Bonnet, F. Vicens, and R. Muñoz, "Zero-touch network slicing through multi-domain transport networks," in *2018 20th International Conference on Transparent Optical Networks (ICTON)*, 2018, pp. 1–4.

[15] P.Alemany *et al.*, "Hybrid network slicing: Composing network slices based on vnfs, cnfs network services," in *IEEE Conference on Network Softwarization (NetSoft)*, 2020.

[16] J. Backman, S. Yrjölä, K. Valtanen, and O. Mämmelä, "Blockchain network slice broker in 5g: Slice leasing in factory of the future use case," in *2017 Internet of Things Business Models, Users, and Networks*, 2017, pp. 1–8.

[17] B. Nour, A. Ksentini, N. Herbaut, P. A. Frangoudis, and H. Moungla, "A blockchain-based network slice broker for 5g services," *IEEE Networking Letters*, vol. 1, no. 3, pp. 99–102, 2019.

[18] "Network functions virtualisation (nfv) release 3; evolution and ecosystem; report on network slicing support with etsi nfv architecture framework," European Telecommunications Standards Institute, Sophia Antipolis, FR, Standard, Dec. 2017.

[19] T. B. G. 2020, "Ganache — truffle suite." [Online]. Available: https://www.trufflesuite.com/ganache