Empirical design, prototyping and evaluation of a new hardware-based network slicing approach for 6G backbone networks
Ricart-Sanchez, Ruben; Salva-Garcia, Pablo; Chirivella-Perez, Enrique; Alcaraz Calero, Jose M.; Wang, Qi

Published in:
2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)

DOI:
10.1109/EuCNC/6GSummit51104.2021.9482495

Published: 28/07/2021

Document Version
Peer reviewed version

Link to publication on the UWS Academic Portal

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the UWS Academic Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
If you believe that this document breaches copyright please contact pure@uws.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

“© © 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”
Empirical design, prototyping and evaluation of a new hardware-based network slicing approach for 6G backbone networks

Ruben Ricart-Sanchez*, Pablo Salva-Garcia*, Enrique Chirivella-Perez*, Jose M. Alcaraz Calero*, Qi Wang*

* University of the West of Scotland, School of Computing, Engineering and Physical Sciences; United Kingdom

Abstract—With 5G being currently deployed worldwide, research into 6G has been initiated to advance the key technologies of 5G to the next level. Network slicing, as a cornerstone technology in 5G, is expected to continue playing an important role in guaranteeing the Quality of Service for various vertical business use cases with divergent yet more demanding requirements in the next generation. This research explores a towards 6G scenario where ultra-high performance of backbone network slicing is entailed as part of an end-to-end solution to address ultra-reliable low-latency communications in mission-critical industrial operations for smart factory and manufacturing, smart warehouse and other Industry 4.0 and beyond applications. To address the highly demanding requirements, this paper proposes a new hardware-based network slicing approach based on the eXpress Data Path (XDP). The proposed solution is designed, prototyped and compared with an existing hardware-based alternative through empirical experiments over a pre-6G infrastructure testbed. The insights gained from the experimentation would help cast light on hardware-accelerated approaches in 6G network slicing.

I. INTRODUCTION

The Beyond Fifth-Generation mobile networks (B5G) and the Sixth-Generation mobile networks (6G) are expected to be increasingly more heterogeneous, denser and more dynamic, requiring stricter performance assurance compared with the current 5G networks [1]. The strict Key Performance Indicators (KPIs) of B5G and 6G networks, together with the large volume of data generated by future connected devices for smart cities, such as industry 4.0 or autonomous driving, will put a strain on communication technologies, which will have to be able to guarantee the required Quality of Service (QoS). In this context, it is paramount to properly accommodate specific service traffic with heterogeneous network requirements to meet the Service-level agreements (SLAs), but also to maximise the utilisation of limited network resources. To this end, network slicing appeared with the objective of providing a programmable, software-driven and holistically end-to-end (E2E) logical communication network that ensures the performance of specific communication services in the current 5G and incoming 6G networks.

There are several benefits of introducing network slicing. Firstly, this technology provides the flexibility required by vertical businesses by creating flexible and dedicated logical networks that meet specific customer demands. Secondly, network slicing reduces the Capital Expenditure (CapEx) of mobile network operators by sharing the network resources within and across network domains. Thirdly, network slices ensures configurable QoS warranties, which provide mobile network operators the possibility of meeting strict and variable SLAs of ultra-reliable low-latency communications (URLLCs). Nevertheless, existing work in 5G have focused on software-based approach in network slicing through virtualisation and softwareisation.

This contribution proposes a new hardware-based backbone network slicing solution for more powerful performance to meet the stricter QoS requirements of 6G use cases. We design and implement a new Smart Network Interface Care (SmartNIC) based scheme to provide network slicing support in the data plane of a towards 6G edge-to-core network segment. The proposed solution is based on an Agilio-Netronome card that allows Extended Berkley Packet Filter (eBPF) programs to be offloaded to it. These eBPF programs are attached to the eXpress Data Path (XDP) available in the SmartNIC. This XDP provides a high performance and programmable network data path available from the Linux kernel version 4.8. Furthermore, this paper compares the proposed XDP-based approach with an alternative hardware-based approach [2], which is a NetFPGA-based network slicing solution. Empirical experiments validate the design of the proposed implementation and yield a set of insightful performance evaluation results.

The rest of the paper is organised as follows. Section II reviews related work on network slicing and SmartNICs implementations. Section III provides an overview of a pre-6G architecture to contextualise the network slicing solution. Section IV presents the technical details of the new XDP-based network slicing solution and compares its main features with an existing scheme. Section V describes the implementation of the proposal and the testbed used to conduct the experimentation. Section VI presents different performance tests and the analysis of the results obtained. Finally, section VII summarises this work and outlines future research.

II. LITERATURE REVIEW

In the recent years, network slicing has become one of the most popular concepts in MEC networks. A network slice is defined by the Next Generation Mobile Networks (NGMN) as “a set of network functions, and resources to run these network functions, forming a complete instantiated logical network to meet certain network characteristics required by the Service Instance(s)” [3]. This network slicing tendency has been followed by several authors and EU research projects, such as 6G Brains [4], which aims to achieve an AI-based End-to-End (E2E) Directional network slicing with guaranteed QoS...
over highly dynamic networks, among other challenges. Also, the EU project SliceNet had the objective of provide a E2E cognitive network slicing and slice management framework in virtualised multi-domain, multi-tenant 5G networks. An overview of the SliceNet framework based on flexible and customise network slicing is presented in [5]. This contribution emphasises on the design and prototyping of an eHealth use case, focusing on the achievement of the E2E QoS-aware network slicing capabilities required, such as low latency and high reliability.

In [6] a NetFPGA-based network slicing implementation for 5G MEC computing architecture is presented, however it does not present support for multi-tenancy and therefore, it is not flexible enough to be compared with this contribution and also to be considered as a 6G solution. In [7], a hardware-based network slicing solution for smart grids self-healing scenarios over 5G networks is described. This smart grid network slicing implementation uses a P4-NetFPGA card and present an empirical validation of the solution proposed. Yan et al. [8] presents a SmartNIC-based solution to enable network slicing for web-scale cloud, with the objective of meeting 5G/Beyond network requirements. This solution uses the P4 language [9] to program the network data plane for L2/L3/L4 parsing and action, achieving very high performance. From the software side, Matencio et al. [10] presents a novel 5G network slicing framework for IoT use cases. This software implementation is able to manage a big heterogeneous IoT network slices dynamically. In [11], Shu and Taleb propose a novel architectural framework of network slicing based on SDN and NFV to guarantee key QoS indicators for different use cases in 5G/B5G networks. Although they present different and very efficient network slicing algorithms implemented in software, the performance results obtained are not sufficient to handle the high volume of traffic expected in 6G networks.

Nevertheless, none of these technologies provides a network slicing solution using a SmartNIC Netronome card and off-loaded XDP. These network slicing capabilities in 6G architectures can be implemented using software components in the kernel stack or hardware elements that can provide better performance benefits. This paper proposes a new hardware-based solution with network traffic processing, filtering and classification performed in the SmartNic.

III. PRE-6G NETWORK ARCHITECTURE

Figure 1 shows a multi-tenant 5G evolution based virtualised architecture that presents the foundations of future 6G architectures, where different verticals and different mobile network operators are sharing the same physical infrastructure. In the Radio Access Network (RAN), the technical development towards 6G to actualise the current 5G use cases (eMBB, URLLC and mMTC) are presented. It is expected that in the incoming generation of mobile networks, new use cases that require extreme performance arise, as well as new combinations of requirements that are not contemplated in current 5G categories [12]. These 5G RANs consist of different antennas and Distributed Units (DUs) connected to Edge segment of the network, where Centralised Units (CUs) are allocated. The CUs bring network services, using Network Function Virtualisation (NFV) technologies, to the last-mile of the network providing better QoE to final users. These CUs also connect the network data plane between the RAN and the Core segment of the network. As depicted, the Edge segment is formed by a physical machine which contains three different NFVs, which act as isolated virtual network operators, sharing the same physical resources.

The Core segment of the network, also shows a physical machine that contains all the different NFVs required by three independent network operators sharing the same physical infrastructure. In this network segment, traffic is received from the different CUs and it is processed in the User Plane Function (UPF), which acts as a mobility anchor for the traffic of the 6G users and also relay the data to internet. The Access and Mobility Functions (AMF)/Session Management Functions (SMF) are used to allocate the IP address and to manage the user sessions on the network. Finally, the Unified Data Management (UDM)/Authentication Server Function (AUSF) are mainly in charge of the user registration, access authorisation, maintaining the network profiles and also, of authenticating servers and proving encryption keys.

The architecture also contains entities for automatic control and management of the infrastructure. The Intelligent Artificial (IA) agent, periodically gathers metrics from the different network elements. These metrics are stored and analysed by this entity, which automatically creates orders that are execute by the management and control entities. On one side, the management plane covers all functions and mechanisms to configure and manage the different tasks and services.
of the infrastructure. On the other side, the control plane generates traffic to control the network so that the required configuration and policies are applied to ensure the expected network behaviour. Both the management and control planes are controlled by the IA, which ensures that the required configuration and policies are inserted on time where they are required.

IV. PROPOSED XDP-BASED NETWORK SLICING APPROACH

As shown in Figure 1, the SmartNic explored in this paper for the proposed new hardware-based network slicing solution is located in the edge-to-core segment of this pre-6G network. In this network segment, the network data plane offloaded in the SmartNIC should support high-demanding 6G virtualised network traffic. These complex data paths are not commonly supported by current Commercial Off-The-Shelf (COTS) cards and therefore, a new hardware-based solution for this scenario is proposed by exploring a Netronome SmartNIC. This solution provides a QoS-aware network slicing offloaded eBPF program into the XDP with multi-tenancy support, which ensures the isolation of the network traffic by user, services or tenant. Each QoS queue used in this solution is directly attached to an isolated CPU of the system, which guarantees the processing isolation of the network traffic. Furthermore, this hardware-based solution allows accelerating network traffic in this network segment, as well as offloading parsing and classification of the data. This SmartNic-based network slicing solution is beyond the state-of-the-art classifiers and the QoS solutions.

For the purpose of comparison, Figure 2 depicts both 4x10GbE NetFPGA-P4 and 2x25GbE Agilio-Netronome approaches, implementing different queuing disciplines for 5G and 6G networks respectively. This figure is divided into three parts: the network adapter, the kernel space and the user space.

As shown on the right side of Figure 2, the Netronome card presents a queuing implementation approach, which is based on a network driver technology that enables the efficient distribution of the network traffic among different queues implemented in the kernel space. The maximum number of queues is directly associated to the number of CPUs in the system. The Netronome card allows attaching eBPF programs offloading them into the XDP hooks available in the hardware device. These programs are written in a restricted C language. eBPF provides diverse key/value data structures called maps. These maps are used to store data that can be accessed from both, the user-space and kernel/hardware-space. When a packet is received on the Netronome and after it is parsed, the different entries should determine the output RSS queue that the network flow should be sent through. The RSS queues allocated in the kernel space are connected to different CPUs, allowing the transmission of the network data traffic through the NFP driver to Linux Networking Stack, the network stack and the sockets that connect the kernel space with the user space. In the user-space two different user applications are presented, the first one to offload the eBPF program/maps and to collect stats, and the second one to configure the offloaded XDP and the maps in the Netronome.

In contrast, the NetFPGA queuing scheme shown on the left side is based on [2], where a network slicing imple-
by 6G networks between users and tenants.

V. IMPLEMENTATION

The complete Netronome-based network slicing scheme and the pre-6G architecture depicted in sections 2 and III, respectively, have been designed and prototyped in our lab facilities. To perform a fair comparison between the NetFPGA solution proposed in [2] and this XDP-RSS Netronome eBPF-based contribution, the same testbed has been used to carry out the empirical validation and comparison. This testbed comprises an edge and a core, which have been deployed in two separate computers with similar hardware specifications: Dell T5810, Intel Xeon CPU E5-2630 v4, 32 768 MB of RAM, and 512-GB solid-state drive (SSD). The pair of Netronome cards developed for this experimentation are Agilio CX 2x25GbE, and they are located in the edge and the core respectively. Furthermore, the Linux kernel version used in edge and core is 5.4.0-050400-generic, with the hyperthreading disabled and ire_affinity adjusted to match one queue per CPU.

Table I shows a set experiments that have been carried out to validate the solution proposed. They are the same experiments used to validate the NetFPGA approach in [2]. These experiments consist of 5 different use cases where depending of the scenario up to 512 with 16 different slices (32 users per slice) are defined and transmitting simultaneously. These experiments envisage two different modes of execution. First, without any eBPF program offloaded in the SmartNIC loaded, with the objective of testing the baseline performance of the card. Second, with the eBPF-based RSS network slicing firmware loaded in the device. The aim is to prove how bare metal network cards are not able to perform RSS queuing when complex network traffic traverses 6G architectures and how our contribution beneficially affects the overall network performance.

VI. EXPERIMENTAL VALIDATION

A. XDP-RSS Netronome vs Netronome Baseline

This section presents the empirical validation of the XDP-RSS Netronome-based network slicing framework proposed. The main aim of the solution is to guarantee the QoS of the users that are within the same slice but also the QoS between different slice instances. To validate this, different parameters, such as delay, jitter and packet loss, have been analysed.

Table I shows a set experiments that have been carried out in order to compare the proposed solution against traditional network card implementations (without 6G traffic support). After this, the same experiments have been carried out, but this time, using our XDP-RSS Netronome network slicing approach. These two set of experiments have been compared in order to see how our solution could improve the network performance.

![Fig. 3. E2E Delay achieved when the number of user per flexible slice is increased by Netronome baseline and XDP-RSS Netronome with 4 RSS queues/CPUs](image)

Figure 3 shows the delay induced by the Netronome without any XDP framework loaded (baseline approach), just a basic L2 forwarding, and the XDP-RSS Netronome network slicing solution implemented, using 4 RSS queues attached to different and isolated CPUs. For these experiments, there are groups of 32 users sending simultaneously 477696 packets of 1500 bytes each group. These groups of users are gradually incremented in groups of 32 per slice and can be formulated as follow:

\[ \sum_{i=0}^{n-1} x_i = 32 \cdot 2^n \]

where \(2^n\) denotes the number of slices defined, in this case a maximum 16. And \(n\) denotes the number of the experiment executed, assuming that \(0 < n \leq 5\).

As shown in figure 3, the average E2E delay applied by the Netronome-baseline approach when 6G traffic is being processed is around 0,045ms, however when the XDP-RSS network slicing implementation is loaded in the card, this delay slightly increases up to 0,06ms. This is due to the fact that the framework implemented is extending the data path card and therefore, increasing the processing time of the network traffic. So, it can be concluded that the extra delay incurred by this implementation is around 0,015ms. Considering that it has been defined that the delay in 6G architectures should be less than 0,1ms [14], the solution proposed in this paper totally meet these requirement.

Figure 4 presents the E2E jitter results in the communication between both network cards, the one in the edge and the one in the core. The baseline jitter of the Netronome has totally meet these requirement. The jitter results obtained fluctuate while the number of user is increased with the patter shown in the form 1. Although there is not a clear difference between the baseline jitter and the XDP-RSS jitter, none of them present a jitter higher than 0,025ms, which meet with the 6G KPIs in terms of jitter for the mMTC use case in warehouses and industry 4.0 [15].

![Fig. 4. E2E Jitter results between the Netronome baseline and XDP-RSS Netronome with 4 RSS queues/CPUs](image)
With regard to packet loss, figure 5 depicts the percentage between the baseline approach and the network slicing approach. This percentage represents the number of packet loss by the Netronome allocated in the core segment of the network. As can be appreciated, the baseline approach obtains higher number of packets lost when users are increased. However, the XDP-RSS network slicing framework does not lose almost any packet while the number of users is increased. Therefore, it demonstrated that the QoS-aware implementation carried out for this contribution improves the reliability of the XDP-RSS Netronome card and made this suitable for 6G scenarios, where the number of packet loss required is almost 0% [14].

As it has been represented, in figures 3, 4 and 5, when the number of users is exponentially increased, the values of delay, jitter and packet loss remain almost constant. These results prove the scalability of the XDP-RSS network slicing solution presented in this paper. The Netronome baseline approach commented in the previous section is also shown in this comparison, however, this approach does not present queuing discipline and therefore, all the traffic is processed by the queue 0 represented in the following graphs. All values shown in these figures have been obtained by following the edge to core E2E test-bed described in section V.

In terms of delay, figure 6 represents a comparison between the E2E delay obtained with the network slicing approach using the NetFPGA and the Netronome card. Notice that in both scenarios the same quantity of network traffic has been transmitted and the same bit-rate has been used. However, it is noticeable how the NetFPGA presents a higher delay in the less priority queues, while the XDP-RSS Netronome presents a delay that is almost constant. Also, even in the highest priority queues of the NetFPGA card, the delay is almost the double than the delay values of the XDP-RSS Netronome. Knowing that the E2E delay expectations in terms of delay in 6G networks are around 0,1ms, the NetFPGA-based network slicing approach does not meet this requirement with the lowest priority queue. Nonetheless, the delay value obtained in the network slicing XDP-RSS Netronome approach presented in this contribution totally meets this value [14].

Figure 7 shown the network performance obtained by the Netronome and the NetFPGA network slicing implementations in terms of E2E jitter. This is a very critical value and it has been demonstrated that 6G networks require extremely low jitter communications, in the order of microseconds [15]. As can be seen in this figure, the higher values of jitter obtained by the NetFPGA implementation are around 2ms in the lowest priority queues. However, the XDP-RSS Netronome solution proposed in this paper, obtain an average of 20µs, which implies up to 100 times better performance than the NetFPGA solution. Furthermore, and contrary to the NetFPGA, the jitter values in the XDP-RSS Netronome remain constant in all the queues. Therefore, it has been demonstrated that the XDP-RSS Netronome solution meets the 6G KPIs in terms of jitter.

In terms of packet loss, figure 8 shown even a higher difference between the NetFPGA and the XDP-RSS Netronome network slicing solution presented in this paper. The Netronome baseline approach still loses a maximum of 2.5% of packets, the XDP-RSS Netronome
These technologies are expected to further improve the proposed solution, in terms of technologies, such as DPDK and AF. The experimental validation has demonstrated the effectiveness of the XDP technology for demanding traffic envisioned for 6G. The approach has a 0% average of packet lost. This demonstrates that the solution proposed in this contribution improves the baseline basic Netronome L2-forwarding implementation and also the NetFPGA network slicing solution presented in [2].

As it has been demonstrated in this section, both implementations show very good scalabilty when the numbers of users is increased. However, there is a substantially high improvement when the Netronome card is used to program the network slicing module. The XDP-RSS Netronome results totally meet the 6G KPIs in terms of delay, jitter and packet loss. Nonetheless, the NetFPGA solution only meets the 5G KPIs and this is why it can not be used in 6G scenarios.

VII. CONCLUSION

Demanding use cases motivated by the forthcoming 6G networks entail new-generation high-performance network slicing technologies to meet the stricter QoS requirements for critical services in Industry 4.0 and beyond. This paper has proposed an architectural and functional design of a new hardware-based network slicing solution built on a Netronome SmartNIC. This solution creates an accelerated network data path based on the XDP technology for demanding traffic envisioned for 6G. The empirical validation has demonstrated the effectiveness of the proposed design and implementation and the achievement of the QoS required by 6G in terms of delay, jitter and packet loss, compared with an existing NetFPGA-based network slicing solution. The evaluation results have shown the superior performance of the Netronome XDP-based approach.

Future work will seek to explore new kernel bypass technologies, such as DPDK and AF-XDP. These technologies are expected to further improve the proposed solution, in terms of packets processing capabilities and throughput. Furthermore, new queuing algorithms will be explored to provide more efficient network slicing implementations for 6G architectures.

ACKNOWLEDGMENT

This work is funded by the European Commission Horizon 2020 5G-PPP Program under Grant Agreement Number H2020-ICT-2020-2/101017226 “6G BRAINS: Bringing Reinforcement learning Into Radio Light Network for Massive Connections” and under Grant Agreement Number H2020-ICT-2020-2/101016941 “5G INDUCE: Open Cooperative 5G Experimentation Platforms for The Industrial Sector NetApps.”

REFERENCES