Thunderbolt-3 Backbone for Augmented 5G Network Slicing in Cloud-Radio Access Networks

Bruno Dzogovic
Oslo Metropolitan University Norway
bruno.dzogovic@oslomet.no

Thanh van Do
Telenor & Oslo Metropolitan University Norway
than-van.do@telenor.com

Bernardo Santos
Oslo Metropolitan University Norway
bersan@oslomet.no

Van Thuan Do
Wolffia AS Norway
vt.do@wolffia.no

Boning Feng
Oslo Metropolitan University Norway
boning.feng@oslomet.no

Niels Jacot
Wolffia AS Norway
n.jacot@wolffia.no

***Abstract* – As the concept of virtualized SDN-enabled cloud radio access networks (C-RANs) paves the way towards the emerging 5G platforms, in parallel, the current available technologies can supplement the progression and open new doors towards establishing a cost-efficient and highly robust 5G infrastructure. According to that, in this work we propose a solution that consists of a Thunderbolt-3 networking between the locally deployed 5G base station and the core network that allows higher capacity and refutes the necessity for fiber networking with more expensive NICs, transceivers and single-mode/multi-mode cables. Analogously, the serving site can accommodate thus more users and enable provisioning of multiple high-bandwidth network slices. In addition, the cost-effectiveness of this approach can allow operators and small businesses, as well as individual users, to deploy own 5G Remote Radio Heads in their own datacenters with distributed antenna systems, consolidating the fronthaul interfaces between the baseband processing and the radio frontend.**

Keywords – 5G, C-RAN, Network Slicing, SR-IOV, Thunderbolt, Container Technology

I. INTRODUCTION

In an era of rapid development of the communication systems, the popularity of 5G gains momentum as the emergence of the Internet of Things (IoT) devices transpire. The focus of the 4G LTE was pointed mainly towards mobile communications and personal/enterprise connectivity worldwide using mobile devices. This context exclusively does not correspond well to the fact that within short period of time, dozens of billions of small devices will join the Internet and require exponentially more resources, as well as connection diversity in terms of protocol and technology.

To circumvent this necessity, 5G promises to establish wider palette of services that are bound to the resource-rich infrastructure. One of the main entities responsible for this corollary is the concept of **Network Slicing** in 5G. According to the 5GPP workgroup, *"The network slice is a composition of adequately configured network functions, network applications, and the underlying cloud infrastructure (physical, virtual or even emulated resources, RAN resources etc.), that are bundled together to meet the requirements of a specific use case, e.g., bandwidth, latency, processing, and resiliency, coupled with a business purpose"* [1]. In other words, network slicing will enable customization of compute, storage and networking functions of the infrastructure for a specific Virtual Network Operator's (VNO) traffic requirements.

Collectively with network slicing, 5G combines other enablers such as Software-Defined Networking (SDN) and Network Function Virtualization (NFV) in order to deliver softwareized virtual solutions into the premises of the operator as well as at the network edge close to the users. For this to ensue, the operators need to divide the functionality of the evolved Node-B (eNB) or in 5G terminology, the next-generation NodeB (gNB) base station into functional splits, where the delivery of radio frontend resources is segregated into Remote Radio Heads (RRHs) and Baseband Units (BBUs), which connect to a local or remote Evolved Packet Core (EPC) 4G network core or 5GC (5G Core). Namely, the physical layer radio function mapping/de-mapping is encoded into Ethernet frames and transported to the baseband unit through the CPRI (Common Public Radio Interface) protocol, defined within the NGFI (Next Generation Fronthaul Interface) working group and the IEEE P1914.1 standard for packet-based fronthaul transport networks [2]. The requirements for a functional split are aiming towards higher bandwidth connection between the RRH and the BBU, while maintaining as lowest possible latency in order to avoid synchronization issues. This inclines towards necessity for fiber networking infrastructure.

Consequently, in this paper, we propose a cost-effective solution for the fronthaul interface at the Oslo Metropolitan University Secure 5G4IoT Lab[1] within the scope of the H2020 CONCORDIA project[2]; in parallel, disregarding whether that is the standard CPRI interface or Radio over Ethernet (RoE) [2], which is based on the Intel Thubderbolt™ technology [3] as well as supplemented by VT-d (Virtualization Technology for Directed I/O) and SR-IOV (Single Root - Input/Output Virtualization) network virtualization techniques [4]. By eliminating the necessity for expensive SFP optical Network Interface Cards (NICs), the Thunderbolt-3 networking should allow for localized deployment of a C-RAN infrastructure and expanded possibility for implementing high-bandwidth network slices. Conclusively, we deliver an

---

[1] http://5g4iot.vlab.cs.hioa.no/

[2] https://www.concordia-h2020.eu/

DOI: https://doi.org/10.1109/5GWF.2019.8911710

experimental evaluation of the performance between the BBU and the RRH endpoints using Thunderbolt-3, providing an insight into the relative values of latency and bandwidth through the virtualized interfaces respectively, as well as pointing-out the advantages of the particular implementation's cost-efficiency.

The paper starts with introducing the necessary components used in the research, as well as the basic 5G functional split architecture. Progressively, the solution encompassing the Thunderbolt-3 technology is elucidated and the implementation demonstrated. We conclude with the adequate findings, pointing out the strengths and weaknesses of using the specific approach.

II. RESEARCH BACKGROUND

A. The C-RAN concept

As stated previously, the C-RAN architecture aims to divest the radio frontend function from the dedicated base station. Another advantage of such setup is cost-efficiency, where the operating costs of a specific base station hardware is drastically cut due to virtualized deployment into a cloud-centric environment. The virtual function of the baseband unit (BBU) can be installed in a local or regional cloud, which doesn't require additional investment for specific hardware. The baseband units thus connect to the remote radio heads (RRH), i.e. antennas at the user's premises or city deployments (see Figure 3).

The flexibility the C-RAN architecture offers is allowing the operators to tweak the parameters for the network so the users experience minimal interference and optimal quality of service. However, one disadvantage of the functional split of the eNB is the high throughput requirement between the RRH and the BBU. The International Telecommunications Union (ITU) defines variety of different split options, based on the functionality of the components for the 5G networks [5]. One example is a functional split of Layer-1 physical functions (RF and Low-level PHY) at the RRH, and further on the High-PHY, MAC, RLC, PDCP and RRC levels to the BBU. This split is known as Option 7. Another popular option is the Option 2, where the RRC and PDCP layers are situated at the Baseband Unit and the RLC, MAC and Physical layers are processed at the RRH. The differences between the 4G and 5G functional splitting are depicted in Figure 1.

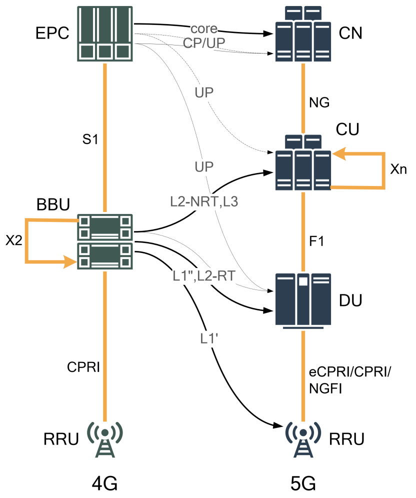


Figure 1. Functional Splits in 5G (courtesy of ITU)[5]

B. Intel Thunderbolt technology

Since the eCPRI/NGFI interface requirements are high in terms of bandwidth, the link between the radio frontend and the centralized processing needs to be robust. The usual solution is the implementation of fiber channel between the components, allowing for low-latency and high speed networking. However, this solution can be expensive and require specialized equipment.

To remedy the situation, we propose usage of the Intel's Thunderbolt-3 technology, aimed for offering higher speed connections between devices with USB Type-C connection interface. Initially developed by Intel, the Thunderbolt interface was invented for transferring video data of high-resolution content, which requires high bitrate speeds, as well as support for external graphics and multiple monitors stacking. The functionality of the interface however, prunes towards instantiating networking interfaces that are having the same function as a typical NIC interface. The capacity of Thunderbolt-3 in terms of speed is 40 Gbps, but the dedicated networking driver marginalizes this to a 10 Gbps fiber-like link. This is expected to be expanded as the protocol diverges towards an open-standard that Intel has released from March 2019, planning a future release of the Thunderbolt protocol merged to as USB 4.0 with a unified USB Type-C connector [3].

C. Network Virtualization and SR-IOV

Virtualization has been accepted as a de-facto method for many datacenters and cloud infrastructures. Although the technologies offered by variety of vendors differ in the approaches they employ to deliver virtualization, in the essence they share many common characteristics. Network virtualization is the method of creating network function resources that are scalable and translate directly onto the networking hardware at which it is engaged [7].

The **Single Root – Input/Output Virtualization** (SR-IOV) specification defines a standardized mechanism to virtualize PCIe devices. This mechanism can virtualize a single PCIe Ethernet controller to appear as multiple PCIe devices. Each device can be directly assigned to an instance, bypassing the hypervisor and virtual switch layer. As a result, users are able to achieve low latency and near-line wire speed. Utilizing the flexibility provided by Software-Defined Networking, traffic can be routed to different Virtualized Network Functions (VNFs) to perform any number of Network Function workloads, such as load balancing, routing, deep packet inspection, etc. As shown in Figure 2, SR-IOV can be an exceptional technology to use for a NFV deployment; expending one or more SR-IOV Virtual Functions (VFs) in a VNF Virtual Machine (VM) or container provides the best performance with the least overhead (by bypassing the hypervisor vSwitch when using SR-IOV) [8].

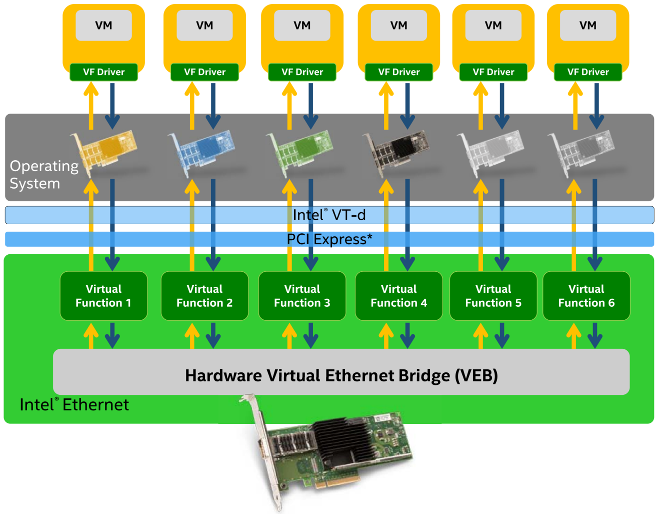


Figure 2. SR-IOV Virtual Network Functions (courtesy of Intel) [8]

D. Container technology

Whether an infrastructure is deployed based on open-source Linux, Windows or any other platform, container technology exists in order to offer immutability and OS-independent instantiations of applications. As containers offer many advantages over virtual machines (VMs), the most important is the lightweight approach of delivering applications as well as microservices architecture [9].

In this work we utilize the Docker [10] container technology in order to deploy a virtualized 5G mobile network function. Namely, a container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. Container images become containers at runtime and in the case of Docker containers - images become containers when they run on Docker Engine. Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging [10][11].

E. 5G network Slicing and FlexRAN controller

One of the most important driving concepts of 5G is the network slicing technique. Unlike Quality of Service (QoS), network slicing aims to provide customized holistic end-to-end virtual network for a specific vertical with tailored service parameters. Quality of Service on the other hand, focuses explicitly on the key parameters of specific service, disregarding the network modeling element [12].

Therefore, in the experiments we take into consideration the network slicing scenario, pertained by the FlexRAN controller for instantiating a personalized Radio Access Network. The FlexRAN platform is made up of two main components: the FlexRAN Service and Control Plane and FlexRAN Application plane. The FlexRAN service and control plane follows a hierarchical design and is composed of a Real-time Controller (RTC) that is connected to a number of underlying RAN runtimes, one for each RAN module (e.g. one for monolithic 4G eNB, or multiple for a disaggregated 4G and 5G) [13].

III. The 5G4IoT Thunderbolt Solution

In the premises of the Oslo Metropolitan University, Norway and the 5G4IoT Lab, we have established a testbed that simulates the enterprise deployment of a 5G network [11], which is based on the **OpenAirInterface** open-source platform [14]. As represented in Figure 3, the eNB/gNB base station is a Software-Defined Radio (SDR) USRP B210 by National Instruments [15]. In a cloud datacenter location, a BBU and RRH instances are deployed in Docker containers, on two different server machines connected through a thunderbolt interface and a private network 10.0.0.0/24. The User Equipment (UE), namely IoT devices and mobile phones connect to the base station using customized SIM cards, for which the user data is registered in the HSS (Home Subscriber Server) database in the core network. The core network is composed of a service and packet gateway that routes the traffic from the main physical interface of the EPC machine in which the Docker container cluster is running. The Docker virtual interface is coupled with the SR-IOV virtual function of the NIC card in the machine. A VPN link is created between the Cloud Gateway and the EPC/5GC machine, which intends to provide a private access to the network core from the devices that are reaching the radio frontend.

To demonstrate a simple network slicing, the FlexRAN controller is used to instantiate two different users with different IMSI values of the SIM cards. Each SIM card corresponds to a separately instantiated Mobility Management Entity (MME) in the core network that assigns the specific user's IMSI to the value in the conforming HSS database to which the MME is performing the DIAMETER authentication. In this case, we also demonstrate an example of a network slice isolation where a single MME can only attribute a specific set of IMSI values to the HSS for which it is allocated. The Container Network

Interface in Docker has thus conventional strict policy to different network subnets for each MME separately, disallowing redistribution of routes and isolating the HSS_1 from HSS_2.

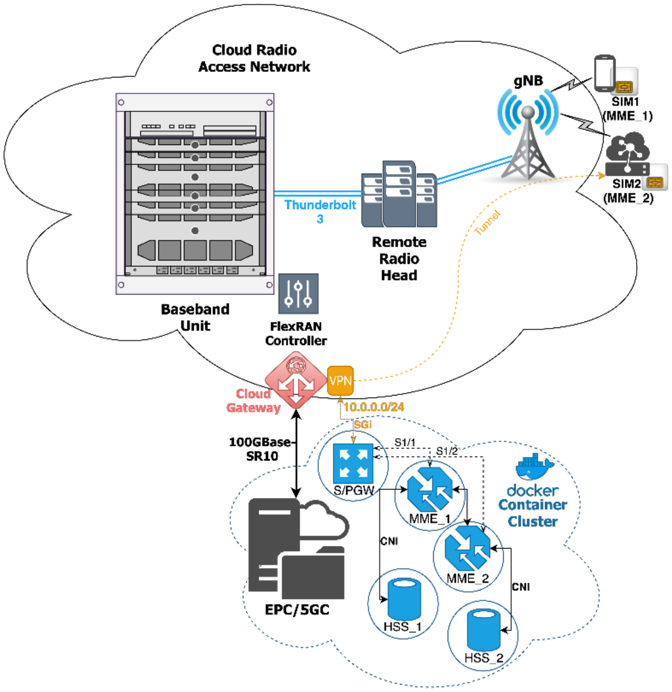


Figure 3. 5G4IoT Lab deployment of a 5G C-RAN network supported by Thunderbolt-3

Conclusively, the exchange of information between the radio frontend and the Baseband Unit is achieved through the two machines connected via Thunderbolt-3 interfaces. At each Thunderbolt endpoint, a SR-IOV virtualization is enabled and four virtual interfaces set (each analogous to 10 Gbps link). Since the Thunderbolt is a bi-directional communication protocol, it allows daisy-chaining of devices. Coupling four virtual links in aggregated mode can allow for more flexible network function virtualization and assigning specific network slices through the virtual network endpoints, as well as integration with the lower-level network fabrics.

IV. IMPLEMENTATION

As an initial testbed, a Thunderbolt-3 PCIe card is installed in one machine, which connects a secondary server with a Thunderbolt-3 2m cable. The Thunderbolt-3 PCIe card routes an output to a USB 3.0 hub for connecting multiple SDR eNB base stations. The latency the PCIe card exhibits due to clock-gating is approximately 1-4 microseconds, which is negligible in terms of data communications and overhead consideration.

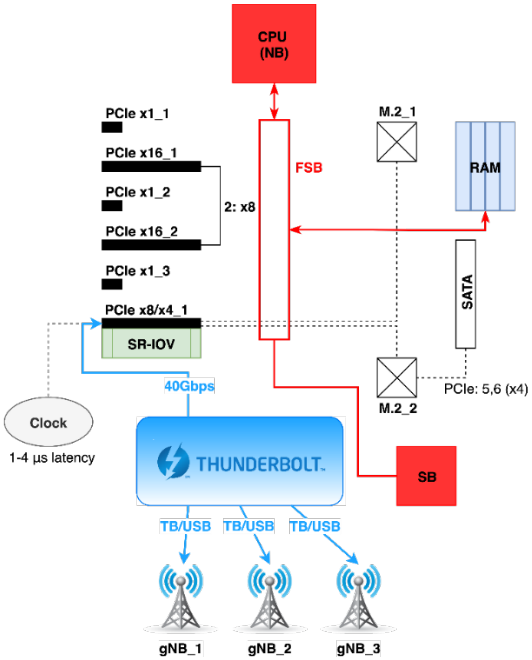
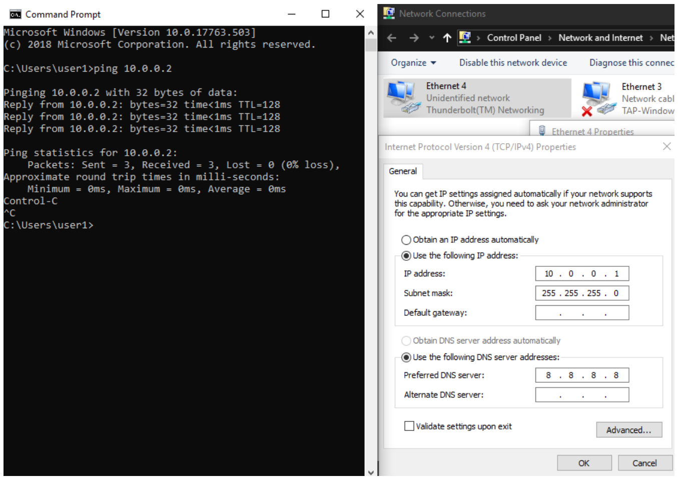


Figure 4. Thunderbolt-3 connection with a PCIe expansion card

In the BIOS, the power-saving states are disabled at PCIe level, for the purpose of ensuring that no fluctuations can impede the transmission process in case of power variations.

To show in parallel a multi-OS deployment, the Thunderbolt-3 endpoints are instantiated on Windows Server hosts, supporting the SR-IOV virtualization with Intel Xeon architecture (Figure 4 and Figure 5). Within this configuration, a Windows Docker container's virtual interface is bridged with the physical NIC of the server. The Docker containers in which the OpenAirInterface components are implemented are built on a Ubuntu Linux 14.04 base image.



```
Microsoft Windows [Version 10.0.17763.503]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\user1>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.2:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C
C:\Users\user1>
```

Ethernet 4
Unidentified network
Thunderbolt(TM) Networking

Ethernet 3
Network cabl
TAP-Window

Internet Protocol Version 4 (TCP/IPv4) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:
IP address: 10 . 0 . 0 . 1
Subnet mask: 255 . 255 . 255 . 0
Default gateway: . . .

○ Obtain DNS server address automatically
● Use the following DNS server addresses:
Preferred DNS server: 8 . 8 . 8 . 8
Alternate DNS server: . . .

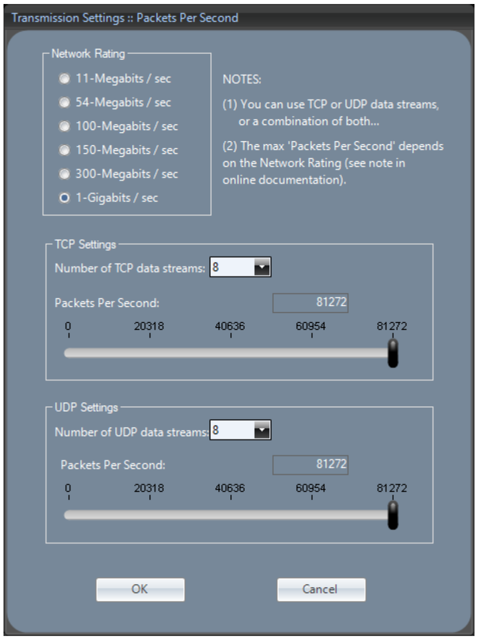
☐ Validate settings upon exit

Figure 5. Windows implementation of a Thunderbolt-3 network function

The direct communication between the virtual functions of the physical hosts allows for an ultra-low latency communication of <1ms latency (Figure *5*). This simply indicates that the two endpoints are connected directly with a single-hop.

V. EVALUATION

To test the connection between the virtual network endpoints of the Baseband Unit and the Remote Radio Head, we establish a 81272 packets per second, 8-stream TCP/UDP communication via the NetStress benchmarking tool (*Figure 6*) [16].

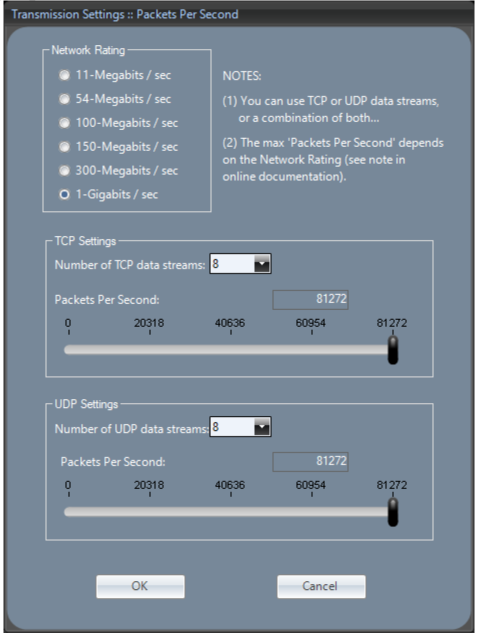

Transmission Settings :: Packets Per Second

Network Rating
○ 11-Megabits / sec
○ 54-Megabits / sec
○ 100-Megabits / sec
○ 150-Megabits / sec
○ 300-Megabits / sec
● 1-Gigabits / sec

NOTES:
(1) You can use TCP or UDP data streams, or a combination of both...
(2) The max 'Packets Per Second' depends on the Network Rating (see note in online documentation).

TCP Settings
Number of TCP data streams: 8
Packets Per Second: 81272

UDP Settings
Number of UDP data streams: 8
Packets Per Second: 81272


Transmission Settings :: Packets Per Second

Network Rating
○ 11-Megabits / sec
○ 54-Megabits / sec
○ 100-Megabits / sec
○ 150-Megabits / sec
○ 300-Megabits / sec
● 1-Gigabits / sec

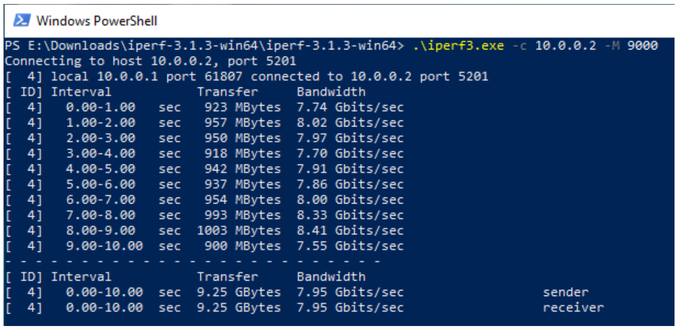
NOTES:
(1) You can use TCP or UDP data streams, or a combination of both...
(2) The max 'Packets Per Second' depends on the Network Rating (see note in online documentation).

TCP Settings
Number of TCP data streams: 8
Packets Per Second: 81272

UDP Settings
Number of UDP data streams: 8
Packets Per Second: 81272

Figure 6. TCP and UDP settings for transmission rate

The capacity of the link is tested with the Iperf3 tool [17] for bandwidth measurements, where the average rate reaches approximately 7.9 Gbits/s. The length of the Thunderbolt cable of 2 meters may attribute to slight signal attenuation, and shortening the overall length can improve the rates and decrease signal absorption (Figure 7).


Windows PowerShell

```
PS E:\Downloads\iperf-3.1.3-win64\iperf-3.1.3-win64> .\iperf3.exe -c 10.0.0.2 -M 9000
Connecting to host 10.0.0.2, port 5201
[  4] local 10.0.0.1 port 61807 connected to 10.0.0.2 port 5201
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-1.00   sec   923 MBytes  7.74 Gbits/sec
[  4]   1.00-2.00   sec   957 MBytes  8.02 Gbits/sec
[  4]   2.00-3.00   sec   950 MBytes  7.97 Gbits/sec
[  4]   3.00-4.00   sec   918 MBytes  7.70 Gbits/sec
[  4]   4.00-5.00   sec   942 MBytes  7.91 Gbits/sec
[  4]   5.00-6.00   sec   937 MBytes  7.86 Gbits/sec
[  4]   6.00-7.00   sec   954 MBytes  8.00 Gbits/sec
[  4]   7.00-8.00   sec   993 MBytes  8.33 Gbits/sec
[  4]   8.00-9.00   sec  1003 MBytes  8.41 Gbits/sec
[  4]   9.00-10.00  sec   900 MBytes  7.55 Gbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-10.00  sec  9.25 GBytes  7.95 Gbits/sec                  sender
[  4]   0.00-10.00  sec  9.25 GBytes  7.95 Gbits/sec                  receiver
```

Figure 7. TCP throughput per-virtual interface

Each virtual channel has a possibility to run simultaneously multiple TCP/UDP streams (referring to the SCTP protocol, which is adopted into the C-RAN architecture). With the same transmission rate of 81272 packets per second, 8 TCP and 8 UDP parallel streams, and MTU set to 9000 Kb jumbo frames, the average performance over time of one minute data transfer reaches 4030.59 Mbps, with standard deviation of +/- 387.08 Mbps for TCP and 3950.19 Mbps with standard deviation of +/- 391.7 Mbps for UDP, respectively. The differences between the rates of the two protocols are hence negligible.

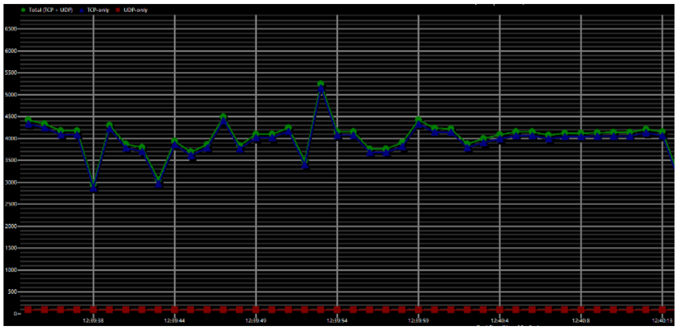


Figure 8. TCP/UDP throughput per-stream

Conclusively, the 8 TCP parallel streams multiplied by the transmission rate of 4030.59 Mbps, yields a link of 32244.72 Mbps, which corresponds to 32 Gbps in total. If the results are compared with the previous Iperf3 tests for bandwidth per-virtual interface that match to 7.9 Gbps, multiplied by 4 virtual interfaces and equals 31.6 Gbps, the similarity is evident and is converging towards the maximum capacity of a Thunderbolt-3 connection of 40 Gbps.

DISCUSSION

Using SR-IOV with Docker containers requires fine-tuning and careful regulation. The process of integrating the SR-IOV virtual function into the Docker Container Networking Interface can prove delicate to manually achieve and may require automated approach. Therefore, if the virtualization is not carefully planned in terms of link capacity, number of virtual interfaces and the possibility of the physical NIC interface, then the potential advantages of SR-IOV can be negated.

Moreover, the maximal Thunderbolt cable length for achieving 40Gbps is disadvantageous compared to the conventional fiber networking, which is only few meters equated to the possible hundreds of kilometers. Accordingly, the idea of C-RAN for long-range remote deployments is not possible by means of the Thunderbolt approach, however the latency for the FFT/IFFT functions transmission between the radio frontend and the Baseband Unit will be incontestable.

The transition of the Thunderbolt technology to an open domain has been released by Intel within the time of writing of this paper, and therefore an expansion and further improvements are expected on multi-platform hardware. If the adoption rate of this technology increases, then it will be feasible to further explore possibilities for implementations in variety of 5G utilizations.

CONCLUSION

Within the 5G4IoT lab, the implementation of a Thunderbolt interface between the 5G radio frontend and the baseband processing has been successfully implemented. As a result, the approach offers possibilities for a customizable network slicing for high-bandwidth and low-latency implementations. Such use cases can prove useful for businesses or individuals who intend to provision a self-hosted 4G/5G virtualized radio frontend and baseband processing, namely for small cell or femtocell deployments in the premises of the organization. The Thunderbolt technology is available on most generic computer hardware and thus doesn't require additional investments in terms of networking equipment. By virtualization of the previously-mentioned network functions, the operators can cost-efficiently plan an alternative way of designing a localized 5G C-RAN infrastructure.

ACKNOWLEDGMENT

This paper is a result of the H2020 CONCORDIA project (https://www.concordia-h2020.eu) which has received funding from the EU H2020 programme under grant agreement No 830927. The CONCORDIA consortium includes 23 partners from industry and other organizations such as Telenor, Telefonica, Telecom Italia, Ericsson, Siemens, BMW, Airbus, etc. and 23 partners from academia such as CODE, University of Twente, OsloMet University, etc.

REFERENCES

[1] 5GPP Architecture Working Group, *"View on 5G Architecture,"* January, 2018. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2018/01/5G-PPP-5G-Architecture-White-Paper-Jan-2018-v2.0.pdf. (Accessed May, 2019).

[2] IEEE Standards Association, *"P1914.1 - IEEE Draft Standard for Packet-based Fronthaul Transport Networks,"* 2018. [Online]. Available: https://standards.ieee.org/project/1914_1.html. (Accessed May, 2019).

[3] Intel Corporation, *"Thunderbolt Technology,"* 2019. [Online]. Available at: https://www.intel.com/content/www/us/en/io/thunderbolt/thunderbolt-technology-general.html. (Accessed May, 2019).

[4] Intel Corporation, *"Virtualization Technology,"* 2019. [Online]. Available at: https://www.intel.com/content/www/us/en/virtualization/virtualization-technology/intel-virtualization-technology.html. (Accessed May, 2019).

[5] International Telecommunications Union, *"Transport Network Support of IMT-2020/5G,"* Technical Report, 2018. [Online]. Available at: https://www.itu.int/dms_pub/itu-t/opb/tut/T-TUT-HOME-2018-PDF-E.pdf. (Accessed May, 2019).

[6] Intel Corporation, *"Intel Takes Steps to Enable Thunderbolt 3 Everywhere, Releases Protocol."* [Online]. Available at: https://newsroom.intel.com/news/intel-takes-steps-enable-thunderbolt-3-everywhere-releases-protocol/#gs.dsnzdp. (Accessed May, 2019).

[7] Y. Zhang, *"Network Function Virtualization: Concepts and Applicability in 5G Networks,"* Wiley-IEEE Press, 2018, p. 9. ISBN: 978-1-119-39060-2.

[8] Intel Corporation Networking Division, *"SR-IOV for NFV Solutions: Practical Considerations and Thoughts,"* February, 2017. [Online]. Available at: https://www.intel.com/content/dam/www/public/us/en/documents/technology-briefs/sr-iov-nfv-tech-brief.pdf. (Accessed May, 2019).

[9] S. Newman, *"Building Microservices: Designing Fine-Grained Systems."* O'Riley, 2015. ISBN: 9781491950340.

[10] Docker Inc., *"Docker Container Technology,"* 2019. [Online]. Available at: https://www.docker.com/. (Accessed May, 2019).

[11] B. Dzogovic, V. T. Do, B. Feng and T. van Do, *"Building virtualized 5G networks using open source software,"* 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, 2018, pp. 360-366. doi: 10.1109/ISCAIE.2018.8405499.

[12] 5GPPP work group, *"5G Network Slicing"*. [Online]. Available at: https://5g-ppp.eu/5g-network-slicing-in-5gtango/. (Accessed May, 2019).

[13] FlexRAN Open-Source Project, *"Flexible and Programmable Platform for Software-Defined Radio Access Networks."* [Online]. Available at: http://mosaic-5g.io/flexran/. (Accessed May, 2019).

[14] OpenAirInterface, *"OpenAirInterface 5G Software Alliance for Democratising Wireless Innovation."* [Online]. Available at: https://www.openairinterface.org/. (Accessed May, 2019).

[15] Ettus Research, *"USRP B210 SDR,"* 2019. [Online]. Available at: http://www.ettus.com/all-products/UB210-KIT/. (Accessed May, 2019).

[16] NutsAboutNets "NetStress Benchmarking Tool." [Online]. Available at: http://nutsaboutnets.com/netstress/. (Accessed May, 2019).

[17] Iperf3, *"Iperf3 Network performance measurement tool."* [Online]. Available at: https://iperf.fr/. (Accessed May, 2019).