This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3108423, IEEE Access

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

5G Campus Networks: A First Measurement Study

JUSTUS RISCHKE[1,4], PETER SOSSALLA[1,5], SEBASTIAN ITTING[1,2], FRANK H. P. FITZEK[1,2] and MARTIN REISSLEIN[3,2]

[1]Deutsche Telekom Chair, 5G Lab Germany, Technische Universität Dresden, 01062 Dresden, Germany
[2]Centre for Tactile Internet with Human-in-the-Loop (CeTI), Technische Universität Dresden, 01062 Dresden, Germany
[3]School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ 85287-5706 USA
[4]Dr. Ing. h.c. F. Porsche AG, Stuttgart, Germany
[5]Audi AG, Ingolstadt, Germany

Corresponding author: Martin Reisslein (e-mail: reisslein@asu.edu).

This work has been supported by the German Federal Ministry of Education and Research (BMBF) as part of the project 5G Insel under the grant 16KIS0956K and by the German Research Foundation (DFG) as part of Germany’s Excellence Strategy EXC 2050/1 – Project ID390696704 – Cluster of Excellence Centre for Tactile Internet with Human-in-the-Loop (CeTI) of Technical University of Dresden.

ABSTRACT A 5G campus network is a 5G network for the users affiliated with the campus organization, e.g., an industrial campus, covering a prescribed geographical area. A 5G campus network can operate as a so-called 5G non-stand-alone (NSA) network (which requires 4G Long-Term Evolution (LTE) spectrum access) or as a 5G standalone (SA) network (without 4G LTE spectrum access). 5G campus networks are envisioned to enable new use cases, which require cyclic delay-sensitive industrial communication, such as robot control. We design a rigorous testbed for measuring the one-way packet delays between a 5G end device via a radio access network (RAN) to a packet core with sub-microsecond precision as well as for measuring the packet core delay with nanosecond precision. With our testbed design, we conduct detailed measurements of the one-way download (downstream, i.e., core to end device) as well as one-way upload (upstream, i.e., end device to core) packet delays and losses for both 5G standalone (SA) and 5G non-standalone (NSA) hardware and network operation. We also measure the corresponding 5G SA and 5G NSA packet core processing delays for download and upload. We find that typically 95% of the SA download packet delays are in the range from 4–10 ms, indicating a fairly wide spread of the packet delays. Also, existing packet core implementations regularly incur packet processing latencies up to 0.4 ms, with outliers above one millisecond. Our measurement results inform the further development and refinement of 5G SA and 5G NSA campus networks for industrial use cases. We make the measurement data traces publicly available as the IEEE DataPort 5G Campus Networks: Measurement Traces dataset (DOI 10.21227/xe3c-e968).

INDEX TERMS 5G measurements, Core delay, Delay variation, Packet latency, Packet loss, One-way delay, Stand-alone (SA) network.

I. INTRODUCTION

A. MOTIVATION 1: NEW 5G CAMPUS NETWORKS FOR INDUSTRY

Numerous emerging technological paradigms, such as Industry 4.0 [1]–[3], Internet of Things (IoT) [4]–[6], and self-driving vehicles [7]–[10], require reliable low-latency communication that is untethered from cables [11], [12]. Proponents of these new technological paradigms have often deferred the provisioning of these required reliable low-latency wireless communication services to the 5th Generation (5G) mobile communication standard developed by the Third Generation Partnership Project (3GPP) [13]–[15]. 5G has been proclaimed as “all-in-one” communications solution for a wide range of application scenarios with stringent requirements for reliable real-time delivery of data packets.

Privacy and security concerns in industrial production plants often necessitate local (private) communication networks for an industrial campus without connectivity to the Internet at large [16]–[19]. In order to support such private campus networks, the 3PGPP 5G specifications include a 5G

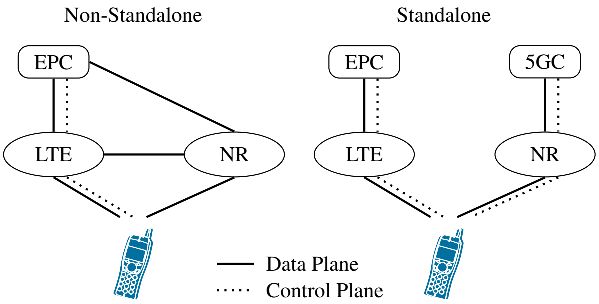


FIGURE 1: Illustration of Non-Standalone (NSA) and Standalone (SA) 5G network structure according to 3GPP specification 21.915 [20, Figure 5.3.2-1].

campus network (non-public network) that is dedicated to a specific organization, e.g., corporation, over the geographic scope of a campus area.

As illustrated in Figure 1, a 5G campus network can operate as a non-standalone (NSA) network to connect a user equipment (end device) via a control plane running over a legacy 4G Long-Term Evolution (LTE) base station and 4G LTE spectrum access (so-called 4G anchor band), while the data plane is provided by a 5G New Radio (NR) base station and a 4G enhanced packet core (EPC). In contrast, a 5G standalone (SA) campus network operates exclusively over the 5G NR base station and 5G packet core (5GC), and does not require any 4G LTE base station, nor 4G LTE spectrum access. 5G campus networks operate over specifically designated frequency bands, e.g., 3.7–3.8 MHz in Germany [21] or the Citizens Broadband Radio Service (CBRS) (3550–3700 MHz) in the USA [22].

B. MOTIVATION 2: LACK OF PACKET-LEVEL MEASUREMENTS FOR 5G CAMPUS NETWORKS

Generally, measurements conducted in testbeds built from prototype equipment are a critical step in the research and development of complex engineering systems, such as 5G communication systems [23]–[27]. To date, 5G testbed measurements have primarily focused on characteristics of the physical layer of the 5G communication system, such as 5G electromagnetic field exposure [28], 5G radio coverage evaluation [29], and similar 5G physical layer aspects [30]–[37]. Another set of testbed studies have focused on 5G aspects related to multi-access edge computing (MEC), i.e., the paradigm of integrating computing capabilities into the 5G communication network infrastructure and operation, e.g., for in-network computation processing [38]–[40] and in-network re-coding of communication data packets [41]–[43]. Specifically, the studies [44]–[47] have conducted evaluations of MEC related frameworks and tests in the context of 5G communication systems.

Furthermore, there has been multiple efforts recently to design and develop large-scale testbeds that would allow for experimentation with a wide range of use cases and user communities. European efforts include the testbed designs [48]–[52], Asian efforts include [25], while related efforts in the United States encompass [53]–[56].

However, the packet-level performance characteristics, e.g., packet latencies (delays) and packet loss probabilities, have not been examined in detail for 5G campus networks. These packet-level characteristics have been examined for 3G and 4G networks in [57], for a public 5G network in [58], and for a 5G NSA network in [59]. More specifically, the study [58] mainly measured round-trip delay times to servers on the public Internet. As the measurements in [58] were conducted on a public network, it was not possible to access the individual 5G network components. In contrast, we conducted measurements on a private 5G campus network testbed with access to all components and could thus measure in isolation the one-way download delay and one-way upload delay with high precision as well as measure in isolation the radio access network (RAN) and core packet losses. The study reported in the conference paper [59] measured an NSA network in a laboratory setting for a fairly narrow set of packet rates and small sets of only 100 packets for a give scenario. In contrast, we consider a wide range of packet rates up to 100000 packets/s and let each scenario run for 1000 s so as to ensure stable results. Overall, in contrast to these prior studies, our present study focuses on *private* 5G campus networks, whereby we consider *both SA and NSA* campus network structures.

C. CONTRIBUTION OF THIS STUDY

As the review of the related testbed studies in the preceding section indicates, the actual performance characteristics of 5G SA and 5G NSA campus networks in terms of the data packet delays and losses in real systems are presently unknown. In this article, we address this knowledge gap by reporting on our development of a rigorous flexible measurement testbed to evaluate 5G SA and 5G NSA campus network packet communication. Our testbed architecture consists of real 5G SA and 5G NSA hardware, namely end devices, radio access network (consisting of antennas and Baseband Units (BBUs)), and packet cores. Our measurement methodology involves a dedicated traffic generation and capture node with connectivity to the end devices and packet core to enable one-way packet delay measurements with sub-microsecond precision. Through packet mirroring at an intermediate switch, we are able to separately measure the processing delay in the packet cores with nanosecond precision.

Our extensive measurement campaigns indicate that the packet delays exhibit relatively wide ranges. 95% of the SA download packet delays are in the 4–10 ms range, while the 95% percentile of the NSA download packet delays reaches 20 ms. Also, both the SA and NSA upload delays reach on the order of 20 ms for moderately high packet rates. In addition, the packet core processing alone can account for large delays that can exceed one millisecond. Moreover, we found that packet losses on the wireless channel are relatively rare for low to moderate traffic bit rates. However, for high packet rates, the packet cores can cause relatively high packet loss probabilities on the order of 0.1% for download core processing and on the order of 10% for upload core processing. Overall, download and upload delays and packet loss probabilities were found to differ substantially.

Our measurements shed light on the actual data packet communication capabilities of 5G Release 15 and can thus inform the ongoing research and development efforts for 5G SA and 5G NSA components. We believe that these measurements with the currently available prototype 5G SA and 5G NSA equipment is especially important as insights gained with the currently available equipment can guide the development and engineering of future generations of 5G SA and 5G NSA equipment that targets industrial use cases. For instance, consistency of the packet delay and improved real-time core packet processing should be priorities for enabling industrial control applications on 5G campus networks. In order to facilitate the dissemination of our measurement results and methodology, we have made our packet measurement traces and code publicly available as IEEE DataPort 5G Campus Networks: Measurement Traces dataset (DOI 10.21227/xe3c-e968 [60]) and the source code is available on GitHub [61].

II. MEASUREMENT SETUP

A. TESTBED

The measurements were conducted with the testbed setup depicted in Fig. 2. The testbed consists of actual 5G network equipment, namely for each network type (SA and NSA), a packet core, BBU, and an antenna. For the packet core, we use in the case of SA the *Open5GS* (Version 2.2.1) [62] open-source 5G packet core and for NSA a proprietary packet core by *Nokia*. For the Radio Access Network (RAN), which consists of BBU and an antenna, we use equipment by *Nokia*. The BBU used is *Nokia* ABIA and ASIA hardware for the 4G part, as well as *Nokia* ABIL and ASIK hardware for the 5G part. As antennas, we use the *Nokia* AirScale Indoor Radio 4G-pRRH AHFHIA and *Nokia* ASiR 5G-pRRH AWHQB units. The SA and NSA cores and RAN are connected via a ICX 7850 switch by *Ruckus*. As end devices we use a *Nokia* FastMile 5G Gateway for NSA measurements and for SA we use the SKM-5xE Router by *Wistron*.

The radio transmits in the n78 band from 3.7 Ghz to 3.8 Ghz with a bandwidth of 100 MHz and a transmitter power output of 17 dBm. The radio operates in the Time Division Duplex (TDD) mode with slot format 1/4 and semi static. The RAN operates with a basic configuration without aggregation and without encryption.

We conducted separate measurements for download (→ in Figure 2), i.e., from the core (to which the traffic generator is attached via a wired connection) to the end device, and for upload (←), i.e., from the end device (to which the traffic generator is attached with a wired connection) to the core (to which the traffic capture node is attached with a wired connection). Each single measurement had a duration of 1000 s. We note that the 1.1 Gbit/s downlink capacity is higher than the 100 Mbit/s uplink capacity. Hence, some of the very high packet rates that are feasible for the download, may be infeasible for the upload (or lead to commensurately high packet losses). In our 5G campus network, the data packets do not have to traverse a public network. Regular public 5G networks would require the data packets to take detours via public network domains, which would add delay.

TABLE 1: Software and hardware versions of utilized end devices.

End Device	WNC SKM-5xE
Software version	v00.13.00.02_01232021022914_perf
Modem version	UMC-A15QE_v13.03
Hardware version	v07.07

End Device	Nokia FastMile 5G Gateway
Software version	3TG00118ABAF77
Chipset	BCM6836 SM8150 SDX50M
Hardware version	3TG00076AAAA

TABLE 2: Radio characteristics of our 4G LTE cell and 5GNR cell as measured by the end devices

5GNR		LTE	
Freq.	3.7–3.8 GHz	Freq.	2.66–2.665 GHz
SNR	20 dB	SNR	30 dB
RSRP	−68 dBm	RSRP	−68 dBm
		RSRQ	−7 dBm

B. DATA TRAFFIC GENERATION AND MEASUREMENT

1) Traffic Generation and Transmission

Accurate measurements require accurate traffic generation and measurement (capture). We selected the MoonGen [63] software for traffic generation and capture as MoonGen can use commodity Network Interface Cards (NICs) with Data Plane Development Kit (DPDK) support and still achieves high-precision packet timestamping and generation of various packet rates. The traffic generator runs a commodity PC with a Intel Core i7-9700 processor, 16GB RAM, and an Ubuntu 20.04 operating system. In addition, the PC is equipped with two Intel x550 NICs. These NICs are capable of generating a timestamp for every incoming packet with nanosecond precision [64].

We generate Constant Bit Rate (CBR) traffic which is predominantly used by automation use cases [12]. We focus on the packet delay, and, more specifically, on the packet One-Way Delay (OWD). For measuring the OWD, timestamps must be captured when the packet is sent (TX) and when it is received (RX) in a given direction, see Fig 2: In the download direction, the packet is sent from the traffic generator via the packet core and RAN to the end device (with directly wired-attached traffic capture); in the upload direction, the packet is sent from the end device (whereby the traffic generator supplies the packet via a direct wire to the end device) via the RAN and packet core to the traffic capture node. The OWD can then be evaluated as the difference between these two timestamps with $\Delta T_{\mathrm{OWD}} = T_{\mathrm{RX}} - T_{\mathrm{TX}}$.

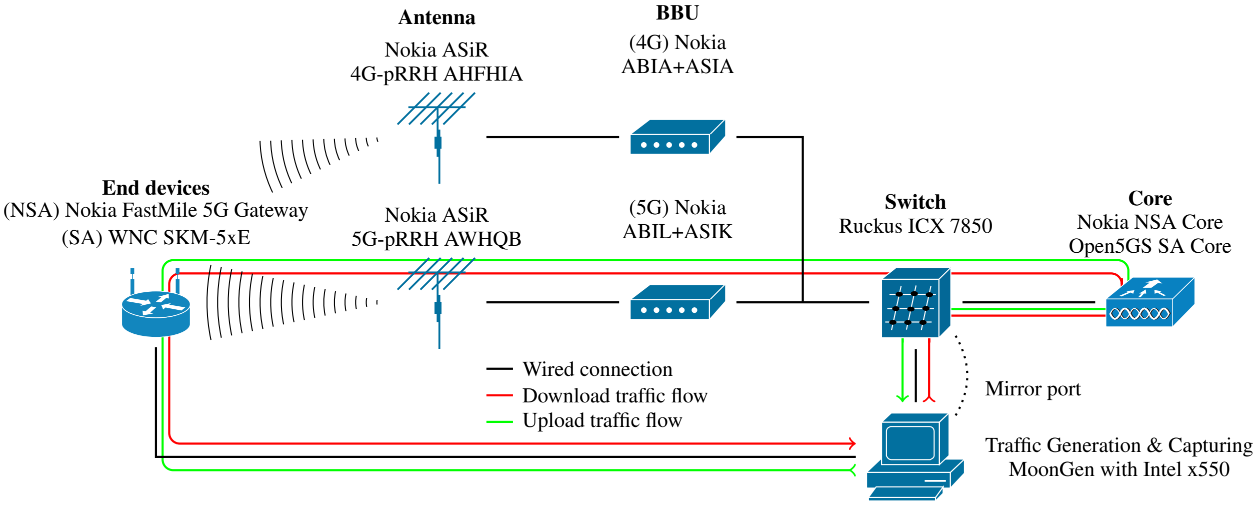


FIGURE 2: Illustration of measurement testbed for the NSA and SA campus network structures of Fig. 1: The 4G LTE radio access network (RAN) is composed of the 4G antenna and 4G BBU, which are utilized for the 5G NSA network. The 5G NR RAN consists of the 5G antenna and 5G BBU, which are utilized for the 5G SA network. The Nokia NSA Core implements the 4G EPC functionality, while the Open5GS SA core implements the 5GC functionality. The download traffic flow emanates from the traffic generator node and traverses the 5G network in the core to end device direction, whereby the packet traffic received by the end device is captured by the traffic generation and capturing node. In contrast, the upload traffic traverses the 5G network in the end device to core direction.

IP, UDP, GTP	UDP	TX Seconds	TX NanoSec.	Sequence Number	Padding

GTP encapsulation ⟷ Measurement packet

FIGURE 3: Measurement packet structure.

The TX timestamp is inserted into each measurement packet by MoonGen using the `clock_gettime()` method to acquire `CLOCK_REALTIME` of the operating system. This approach has a precision of finer than $1\mu s$, when the framework support of DPDK is used [65]. Figure 3 depicts such a measurement packet containing the seconds and nanoseconds of the TX timestamp separately. In addition, there is a sequence number to identify each single packet and calculate its OWD.

2) Traffic Capture

At the receiving side, MoonGen is used as well. The RX timestamps are gathered when packets are captured using again the `clock_gettime()` method. Both the packet TX and packet RX occur at the Traffic Generator machine, i.e., the timestamps are obtained via `clock_gettime()` from the same machine clock, which makes synchronization of multiple machines unnecessary.

The packets together with the RX timestamps are then stored in Packet Capture (PCAP) files for later evaluation. Although the timestamp creation is software-based, the precision is finer than 1 μs [65]. Hardware-based timestamping is possible as well; however, the packet generation rate would be quite limited. In addition, the hardware clocks of the NICs would need to be synchronized, even on the same PC. Therefore, we use software-based timestamping for packet generation. And, to avoid complex time synchronization, capturing uses software as well, which achieves a sub-microsecond precision.

3) Core Delay Measurement

In addition to the total end-to-end OWD, we also evaluate the delay caused by the SA and NSA core. The core uses the GPRS Tunneling Protocol (GTP) to transmit the original data packets to the RAN. The original data packets are encapsulated within GTP packets (as end device and core always communicate via GTP). The GTP tunnel between the end device and the core necessitates packet processing, i.e., the GTP encapsulation/decapsulation packet processing and the minimal packet forwarding processing by the Linux network stack, by both the end device and the core. Since the packets traverse the core, they traverse the switch two times (see Figure 2).

For the download (generator to end device), the packet traverses the switch once before the encapsulation as original measurement packet and once after the core processing. We use the mirror function of the switch to redirect all incoming and outgoing packets to another port to which we have connected the traffic generator (see dotted Mirror port link in Figure 2). The time difference between the time instant when the mirrored copy of the original packet (that traverses the switch in the traffic generator to core direction) arrives to the traffic capture and the time instant when the mirrored copy

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3108423, IEEE Access

of the processed packet (that traverses the switch in the core to end device direction) arrives to the traffic capture gives the core processing delay for the download.

For the upload, the generator sends the original measurement packet to the end device. The end device encapsulates the packet with GTP and sends the GTP-encapsulated packet to the core, which then unpacks the GTP packet before sending it to generator. The time difference between the time instant when the GTP packet arrives from the switch to the traffic generator (the mirrored copy of the traversing end device-to-core GTP-encapsulated packet) and the time instant when the unpacked packet (that has been processed by the core) arrives to the traffic generator gives the upload delay of the core.

Since the processing delay of the packets in the core is on the order of microseconds, the measurement resolution needs to be some orders of magnitude higher. We use a special feature of the Intel x550 NICs, which can timestamp each incoming packet with nanosecond precision [64], whereby the same (given) port on the same (given) NIC is utilized for capturing the core traffic. In particular, to determine the core delay, both the packet traffic entering (going towards) the core and the (core-processed) packet traffic exiting (coming from) the core, is captured on the same NIC port with a timestamp for each packet. Note that for download, the GTP packet exits the core; whereas for upload, the original packet exits the core. Since every measurement packet transmitted by the generator contains a sequence number we can trace the packet (as well as its copies and GTP version) within the 5G system.

4) Link Delay to Traffic Generator and Capture

We acknowledge that the traffic generator to end device connection and the switch to traffic generator connection introduce some delays. The processing, forwarding, transmission, and propagation delays of these two links cannot be avoided in a physical testbed that utilizes 5G bridges/routers. We note that our setup is a minimal approach, hence our measurements indicate the lower delay bounds. Also, compensating for these delays would not be sensible because an end user could not compensate for these delays either. These link delays would only be avoided in scenarios where applications run directly on the core (edge computing) or the end device; however, then other ancillary delays arise, e.g., for the operating system (OS) network stack and for switching packets between VMs. We also note that in our measurement testbed, the delays of these two links are negligible. Specifically the switch to traffic generator link delay for a switch (internal) processing latency of 0.8 μs [66], a maximum packet size of 1280 bytes, and a 10 Gbit/s connection is

$$\begin{aligned} T_{\mathrm{TG,Switch}} &= T_{\mathrm{Proc}} + T_{\mathrm{Trans}} + T_{\mathrm{Prop}} \\ &= 0.8\,\mu\mathrm{s} + \frac{1280 \cdot 8\,\mathrm{bit}}{10\,\mathrm{Gbit/s}} + \frac{2\,\mathrm{m}}{2 \cdot 10^8\,\mathrm{m/s}} \\ &\approx 1.8\,\mu\mathrm{s}. \end{aligned}$$

For the traffic generator to end device link delay calculation, we assume that the end devices are based on Linux, e.g., the Nokia devices are based on *Android*, and use its standard forwarding capability. For a typical forwarding processing delay of 200 μs [67]–[70] in the end device, in conjunction with a maximum packet size of 1280 bytes and a 1 Gbit/s connection,

$$\begin{aligned} T_{\mathrm{TG,EndDev}} &= T_{\mathrm{Processing}} + T_{\mathrm{Trans}} + T_{\mathrm{Prop}} \\ &= 200\,\mu\mathrm{s} + \frac{1280 \cdot 8\,\mathrm{bit}}{1\,\mathrm{Gbit/s}} + \frac{2\,\mathrm{m}}{2 \cdot 10^8\,\mathrm{m/s}} \\ &\approx 210\,\mu\mathrm{s}. \end{aligned}$$

Currently, the data sheets of 5G devices, even for industrial usage, provide no further information on the delay introduced by forwarding the traffic from 5G to Ethernet (for our download), nor the delay introduced by forwarding from Ethernet to 5G (for our upload). We note that in the case of local processing on the end devices, the forwarding delay can be avoided, but the Linux network stack introduces an additional delay of the same magnitude [71].

C. METRICS

1) One-Way Delay

For a given packet, the One-Way Delay (OWD) is evaluated as the difference between the receiving timestamp and the sending timestamp, which are gathered as described in Sections II-B1 and II-B2. The benefit of One-Way Delay (OWD) over traditional Round-Trip Time (RTT)-based measurements, e.g., using *ping*, is that the OWD allows for the detailed investigation of the individual download and upload delay components.

2) Packet Loss

Packet losses can occur in the examined 5G network setup, especially for high packet rates. We generate packets according to a prescribed packet rate, without congestion control. Hence, the data rate can exceed the capacity of the 5G system, leading to packet losses. In the core, packets can be lost because classic socket API programming, as used in Open5GS and presumably also in the Nokia core, is not designed to handle many packets per second of a single stream. In the case of Open5GS, the developers are aware of this issue and modern packet processing frameworks, such as DPDK or Cisco's Vector Packet Processing (VPP), could increase the performance in terms of processing delay and throughput [1]. We denote the core packet loss probability as ϵ_{Core}.

The overall end-to-end (E2E) packet loss probability ϵ_{E2E} over the entire communication path, which consists mainly of the RAN and the core, can be measured in the testbed setup. However, the RAN packet loss probability ϵ_{RAN} cannot be measured directly and needs to be evaluated from the measured ϵ_{E2E} and ϵ_{Core}. From the approximately independent

[1] https://github.com/open5gs/open5gs/issues/759

core packet loss probability ϵ_{Core} and RAN loss probability ϵ_{RAN}, the E2E loss probability can be evaluated as

$$\epsilon_{\text{E2E}} = 1 - (1 - \epsilon_{\text{Core}}) \cdot (1 - \epsilon_{\text{RAN}}),$$

Rearranging, we obtain

$$\epsilon_{\text{RAN}} = \frac{\epsilon_{\text{E2E}} + \epsilon_{\text{Core}}}{\epsilon_{\text{Core}} - 1},$$

which we utilize to deduce where the losses occur in our 5G system.

3) Packet Delay Variation (PDV) and Inter-Packet Delay Variation (IPDV)

In addition to achieving low packet delays, the ability of 5G to deliver packets with a constant delay is important. Hence, the delay variation needs to be evaluated. We consider the two main packet delay variation metrics, namely the Inter-Packet Delay Variation (IPDV) and Packet Delay Variation (PDV) defined according to RFC 5481 [72]. The IPDV is calculated as the difference of the delay between consecutive packets, whereby the average IPDV should always be zero. The PDV is calculated as the difference between each individual packet delay and the minimum packet delay of a measurement interval. In summary, the PDV and IPDV metrics evaluate the variance of the measured packet delays, i.e., assess how “deterministic” the packet delay is on the communication link.

4) Downtime

A downtime measure should characterize whether the communication link can fulfill the delay requirements, e.g., for robot control or not. We consider the number of consecutive packets that exceed a prescribed tolerable maximum delay as downtime measure. In principle, the total number of packets exceeding a prescribed delay requirement within a given measurement interval can already be determined by the Cumulative Distribution Function (CDF) of the OWDs. However, robot control over networks typically requires that no more than 3 to 6 consecutive packets exceed the delay requirement [12].

III. RESULTS: END-TO-END DELAY AND PACKET LOSSES

As first evaluation we examine the total one-way delay OWD and packet losses for the entire (end-to-end) download communication path from the traffic generator (via a wired connection to the packet core) to the end devices as well as the upload communication path from the end devices to the traffic capture (via a wired connection to the packet core).

A. DOWNLOAD

1) Packet Delay: One-Way Delay (OWD)

Figure 4 shows the download OWD for various packet rates for 128 byte packets. We observe from Figure 4 that for a given technology (SA or NSA), the OWD varies in a seemingly counter-intuitive pattern for the different packet rates: Starting from the 10 packet/s rate, the OWD tends to very slightly increase as the packet rate increases to 100 packet/s rate and then the OWD makes a substantial jump as the packet rate is increased to 1000 and 10000 packet/s, while increasing the packet rate to 100000 packets/s substantially reduces the OWD, even slightly below the OWD level for 10 packet/s for SA. This seemingly counter-intuitive OWD behavior appears to be due to batch processing mechanisms in the RAN scheduling and core packet processing. It appears that low packet rates (10–100 packet/s) are processed relatively quickly while moderately high packet rates (1000–10000 packets/s) appear to trigger the creation of larger batches (packet trains) that increase the delays of individual packets. On the other hand, extremely high packet rates (on the order of 100000 packets/s) appear to fill the batches formed for the processing very quickly so that the individual packets are not unduly delayed.

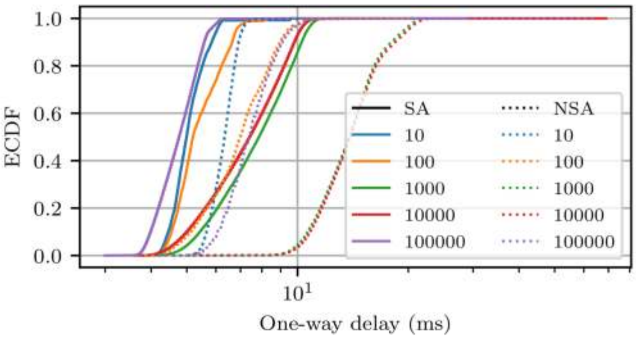


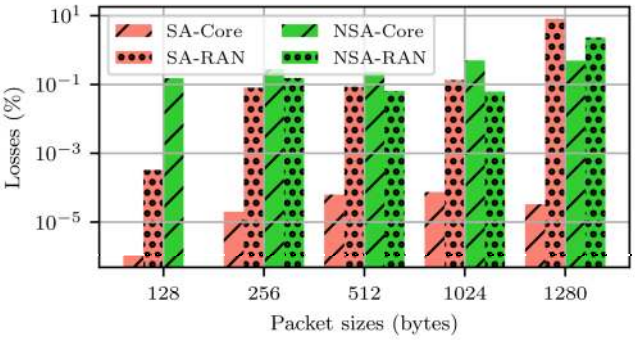
FIGURE 4: Empirical cumulative distribution function (ECDF) of end-2-end download One-Way Delay (OWD) for different packet rates; fixed packet size of 128 bytes.

We also observe from Figure 4 that SA achieved generally approximately 2–8 ms shorter OWD than NSA. This indicates that SA appears generally better suited to achieve OWD below 10 ms.

In additional evaluations for which we do not include detailed plots due to space constraints, we examined the OWD for fixed packet rates as a function of the packet size ranging from 128 bytes to 1280 bytes. We found that the packet size has essentially no effect on the OWD. This consistent with the results in [59] and is plausible since the packet size influences mainly the transmission delay (packet size divided by the transmission bitrate), which is negligible for high data rates, such as in 5G.

2) Packet Loss

We evaluated the packet loss probabilities for the core and RAN in percent (of the total number of transmitted packets) for the packet rates 10, 100, 1000, 10000, and 100000 packets/s for the fixed packet size of 128 bytes. We found that there are no packet losses, except for the high packet rate of 100000 packets per second, which resulted for the SA RAN in $\epsilon_{\text{RAN}} = 3.28 \cdot 10^{-4}\%$ and $\epsilon_{\text{core}} = 10^{-6}\%$ for the



	128	256	512	1024	1280
SA-Core	1.00e-6%	2.00e-5%	6.30e-5%	7.51e-5%	3.30e-5%
SA-RAN	3.28e-4%	8.11e-2%	8.67e-2%	1.36e-1%	8.17e0%
NSA-Core	1.52e-1%	2.79e-1%	2.41e-1%	5.02e-1%	4.95e-1%
NSA-RAN	0.00e0%	1.55e-1%	6.43e-2%	6.23e-2%	2.30e0%

FIGURE 5: Download packet loss probabilities for different packet sizes and a rate of 100000 packets per second.

Open5GS core, while for NSA, $\epsilon_{\text{RAN}} = 0$ and the Nokia core had $\epsilon_{\text{core}} = 1.52 \cdot 10^{-1}\%$. For 100000 packet/s, the required bandwidth is 128 bytes $\cdot\, 8 \cdot 100000 = 102.4$ Mbit/s, i.e., well below the 1.1 Gbit/s download capacity. Hence we can conclude that packet losses can occur in the 5G RAN and core even if the capacity of the air interface is not exceeded. Future research should examine whether cores that employ software or hardware based accelerations frameworks, e.g., [73]–[78], can mitigate the packet losses.

We next evaluate different packets sizes at a packet rate of 100000 packets per second in order to examine the influence of an increasing required packet traffic bandwidth (bitrate). Figure 5 indicates that for the 5G NSA Nokia core, the core packet loss probability ϵ_{core} is rather stable (albeit at a fairly high level on the order of $10^{-1}\%$) as the packet size increases and thus the required bandwidth also increases. In contrast, with Open5GS, there is a core packet loss increase from packet sizes 128 to 1024 bytes, however, the loss probability stays below $10^{-4}\%$. Therefore, we can reasonably conclude that the packet rate is the dominant factor for the packet losses during download in the core.

However, for increasing packet sizes, we can observe in Figure 5 increasing packet losses on the RAN air interface. We observe a dramatic increase of ϵ_{RAN} when the packet size is doubled from 128 bytes to 256 bytes; further packet size doubling to 512 bytes and 1024 bytes tends to only slightly further increase ϵ_{RAN}. Focusing on the large 1280 byte packet size, we observe from Figure 5 that SA suffers mainly losses at the air interface (on the order of 10%) while the SA core losses are negligible (on the order of $10^{-5}\%$). In contrast, the NSA core losses are significant (on the order of $10^{-1}\%$) and only slightly lower than the NSA air interface (RAN) losses.

Overall, the results in Figure 5 thus indicate that the RAN packet losses depend mainly on the utilization of the radio link. Both the NSA and SA radio links have the same bandwidth and should in theory achieve the same throughput. The small differences are likely due to differences of the antenna quality. On the other hand, the core losses depend on the packet rate and the implementation of the respective core technology (SA vs. NSA). A possible explanation for the different behaviors of the SA and NSA cores, is that the 5G NSA Nokia core is a professional solution that offers a larger feature set than the 5G SA Open5GS. The larger feature set could possibly lead to performance reduction in the case of the Nokia core. This causes packets to queue up, resulting in increased core delays, as evaluated in Section IV, and, once queues overflow, to increased losses.

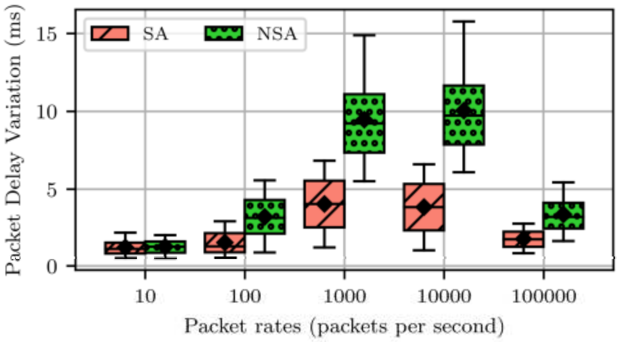


FIGURE 6: End-2-End download Packet Delay Variation (PDV) for different packet rates; fixed packet size of 128 bytes.

3) Delay Variability: PDV and IPDV

Figure 6 shows the PDV for various packets rates for packets with a size of 128 bytes. We observe from Figure 6 that the packet rate has a considerable effect on the PDV (while additional evaluations revealed that the packet size has no significant effect on the PDV). Interestingly, the low packet rates (10–100 packets/s) that exhibited low OWD in Figure 4 give relatively low PDV in Figure 6, while the moderately high packet rates (1000–10000 packets/s) with the high OWD in Figure 4 correspond to relatively high PDV in Figure 6, and the extremely high packet rates (on the order of 100000 packets/s) exhibit relatively low OWD in Figure 4 and low PDV in Figure 6. Also, the generally longer NSA OWD compared to the shorter SA OWD in Figure 4 correspond to higher PDV for NSA than for SA in Figure 6. Thus, overall, the results in Figures 4 and 6 indicate a correspondence between OWD and PDV. Importantly, the relatively high PDV up to 5–15 ms for the moderately high packet rates (1000–10000 packets/s) indicate inconsistencies in the packet delay that could disrupt automatic control that requires consistent periodic updates of the control data every few milliseconds.

Figure 7 shows the CDF of IPDVs. We observe from Figure 7 that for 10, 100, and 10000 packets/s, the IPDV is approximately symmetrically distributed around zero. The maximum inter-packet delay is generally ±1.5 ms for 10

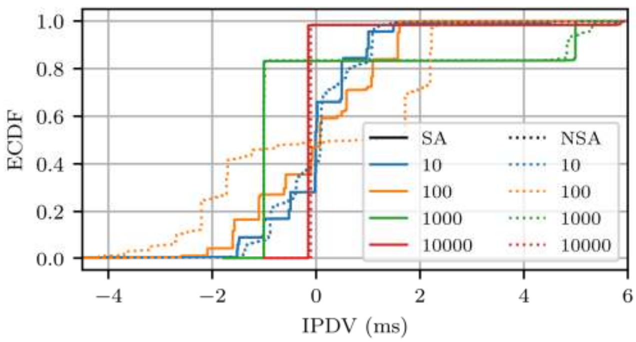


FIGURE 7: End-2-End download Inter-Packet Delay Variation (IPDV) for different packet rates; fixed packet size of 128 bytes.

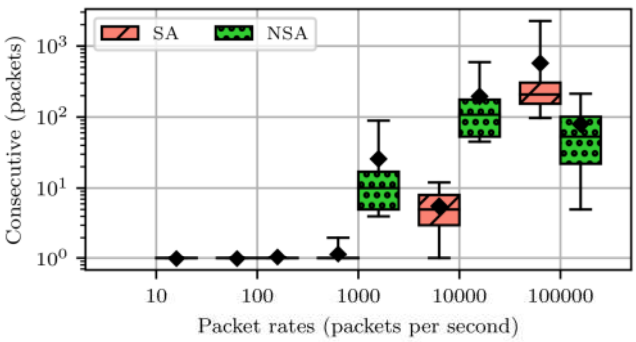


FIGURE 8: End-2-End download: Number of consecutive threshold exceeding packets for different packet rates; fixed packet size of 128 bytes.

packets/s and for SA with 100 packets/s, whereas NSA with 100 packets/s exhibits substantially larger IPDV that reaches from -4 ms to $+2.5$ ms. For 10 to 100 packets/s, the IPDVs occur at multiples of ± 0.5 ms. which is likely due to the scheduling dynamics of the RAN.

We further observe from Figure 7 that for 1000 packets/s, the IPDVs are not symmetrically distributed around zero; rather, the median IPDV is approximately -1 ms. Closer inspection of the packet delay traces revealed an oscillatory saw-tooth-like behavior of the individual successive packet delays. These saw-tooth dynamics result in few relatively large positive inter-packet delays (around $+5$ ms) during a rising edge of a saw-tooth, and numerous relatively small negative inter-packet delays (around -1 ms) during the falling edge. A possible explanation for these dynamics is the slotted medium access in 5G. As long as the channel is reserved for a sender, the packets experience relatively short delays (resulting in the small negative inter-packet delays). When a reservation expires, packets are queued, causing a rapid increase in delay (rising edge of saw-tooth with relatively large positive inter-packet delays). As soon as the channel is reserved again, the queues are emptied, returning the delays to low levels. A possible work-around this issue could be the usage of Frequency Division Duplex (FDD) instead of TDD. However, for the common n78 frequency band used in this study, the usage of TDD is mandatory.

The ratio of the IPDV to the OWD is especially high for low packet rates (10, 100 packets/s). This could be an issue for robot control e.g., via *Profinet*, which usually requires a low delay variation.

4) Downtime

In addition to the delay variation, it is important to know how many consecutive packets exceed a delay requirement, as evaluated by the *Downtime* metric introduced in Section II-C4. We set a 10 ms delay threshold. Figure 8 shows the number of consecutive threshold-exceeding packets for different packet rates for packets of size 128 bytes. Figure 8 indicates that in the range of 10 to 100 packets/s, only at most one consecutive packet exceeds the delay threshold for both SA and NSA (not a single packet exceeded the 10 ms threshold for the SA 10 packet/s scenario; therefore, this scenario does not appear in Figure 8). Starting with 1000 packets/s, the delay generally increases (see Figure 4); accordingly, it is to be expected that the number of consecutive packets that exceed the 10 ms threshold also increases. For 1000 packets/s, SA ensures that more than the probability range represented by the whiskers (from $[Q_1 - 1.5(Q_3 - Q_1),\ Q_3 + 1.5(Q_3 - Q_1)]$ with Q_1 and Q_3 denoting the first and third quartiles, respectively) of the spans of consecutive threshold-exceeding packets are two or less, while for NSA, more than 75% of the spans are twenty or less; thus, robot control, which can have 3 to 6 consecutive packets exceeding the threshold, would still be feasible with SA. Interestingly, for 100000 packets/s, long spans of 100 or more consecutive packets that exceed the delay threshold can occur, despite the very short OWD for the 100000 packets/s rate (see Figure 4). Apparently, the efficient batching at the high packet rates, which achieves short OWD can leave on the order of hundred to a thousand packets at a stretch out of a current processing batch, and thus cause large delays for these unfortunate packets. This underscores the importance of examining and addressing the distribution of packet delays at the upper end of the delay distribution in order to rigorously examine reliability.

Note that the downtime [in units of seconds] during which no update is delivered to a control application is the inverse of the packet rate multiplied by the number of consecutive packets that exceed the prescribed delay threshold. For instance, for 100 packets/s, which has at most one consecutive packet exceeding the delay threshold, the downtime is $(1/100)$ [1/packets/s] × 1 packet $=$ 10 ms, while for 1000 packets/s (and the probability range $[Q_1 - 1.5(Q_3 - Q_1),\ Q_3 + 1.5(Q_3 - Q_1)]$), SA has a corresponding downtime of 2 ms. Thus, one potential strategy for reducing the downtime [in units of seconds] is to effectively increase the packet rate of robot applications with inherently low packets rates through multiple transmissions of each packet, i.e., essentially through repetition coding. For instance, an application with an inherent packet rate of 100 packets/s can reduce the downtime from 10 ms to 2 ms by repeating each packet ten times (thus effectively sending with a packet rate of 1000 packets/s).

5) Summary and Discussion

In summary, losses in 5G mobile communications are generally rare. However, packet losses can occur for high packet rates and are exacerbated by large packet sizes. The classic socket API programming in the currently available core implementations apparently lacks the efficiency for processing high packet rates. In the context of 5G as an industrial communication standard, future research and development needs to shift its focus away from the "few-big-packets" view towards a "many-small-packets" view of communication for industry use cases. The Linux operating system used in end devices and for hosting the core is designed to process few big packets (or batches of small packets), as they are more optimal in terms of overhead (and throughput). However, the 5G wireless network will be used to bridge between wired infrastructure (where the robot control is deployed) and one or more robots.

Robot control is characterized by small periodically sent packets, e.g., with a rate of 1 kHz (i.e., 1000 packets/s). If there are several robots behind a 5G gateway, the required packet rates accumulate and hence can readily lead to aggregated packet rates in the moderately high to extremely high range (10000–100000 packets/s). The necessity and challenge of achieving high packet rates is often neglected, e.g., in [79], where robot control is described with low latency, low data rate (because of small packets), and high reliability, i.e. a classical Ultra-Reliable and Low-Latency Communications (URLLC) use case. However, these rarely considered required packet rates can strongly influence the 5G network performance, as demonstrated by the results in this section.

In addition, the Media Access Control (MAC) must be interfaced in a synchronized, i.e., time-coordinated, manner with the application processes running on the devices that send the packets. Mismatches between the sending processes on the end devices and the channel reservation processes on the 5G network can lead to packet delay variations and varying inter-packet delays which then in turn can lead to packets exceeding delay thresholds. More specifically, widely scattered packet delays result in a wide spread in the ECDFs as depicted in Figure 4, high PDVs as shown in Figure 6, and an asymmetric IPDV distribution as indicated in Figure 7. These delay characteristics are typically not a major problem for throughput-centric applications, such as media delivery with pre-buffering. However, for cyclic real-time communication, these delay characteristics pose critical challenges.

Current research, e.g., in [80]–[85], has begun to tackle the integration of Time-Sensitive Networking (TSN) into 5G systems, to enable real-time communication. However, a main remaining challenge are mechanisms for timely RAN transmissions, e.g., channel pre-allocation mechanisms that ensure that packets are not queued but directly sent. One approach could be to design and operate a time synchronized network, where end devices, 5G RAN, and 5G core are all synchronized. The time synchronization could avoid the accumulation of multiple packets along the network transport path, and thus additional queuing delay and packet losses.

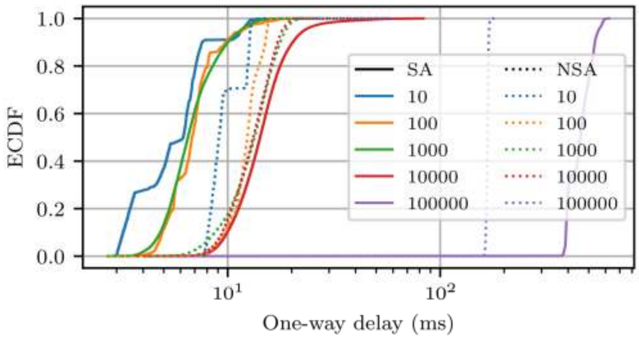


FIGURE 9: Empirical cumulative distribution function (ECDF) of end-2-end upload One-Way Delay (OWD) for different packet rates; fixed packet size of 128 bytes.

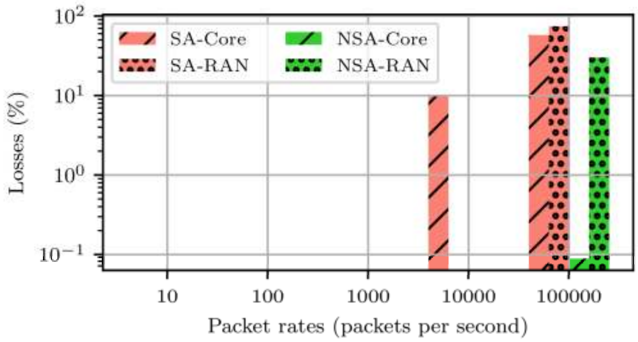
B. UPLOAD

1) One-way Delay

Figure 9 depicts the OWD distribution for various packet rates for a packet size of 128 bytes. For SA, the upload results in Figure 9 are comparable to the download results in Figure 4 for 10 to 1000 packets/s. Starting with 10 000 packets/s, we observe from Figure 9 that the 5G SA network exhibits a pronounced upload delay increase compared to the lower packet rates and the corresponding download delays in Figure 4. In addition, at 100 000 packets/s, the nominal uplink capacity of 100 Mbit/s is exceeded and hence packets are dropped due to congestion.

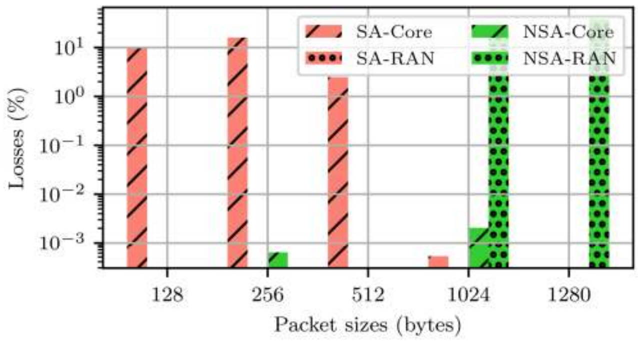
2) Packet Loss

Figure 10 depicts the upload RAN and core packet loss probabilities for a range of packet rates for a fixed 128 byte packet size. For a packet rate of 10000 packets/s, we observe from Figure 10 a packet loss probability of almost 10% at the 5G SA Open5GS core. At 100000 packets/s. the losses caused by the Open5GS core increase to nearly 60%. In addition, there are now also losses in the RAN, namely about 74% for SA and 30% for NSA. Thus, for 100000 packets/s at a packet size of 128 bytes approx. $100000\,\text{packets/s} \cdot 8 \cdot 128\,\text{bytes} \times (1 - 0.744) \approx 26.2\,\text{Mbits/s}$ effectively arrive (GTP overhead not counted) to the core in the SA RAN and about 71.7 Mbits/s in the NSA RAN. However, these losses are not necessarily caused because the capacity of the RAN is exceeded. This can be seen in Figure 11, where the packet loss probabilities for various packet sizes are depicted for a packet rate of 10000 packets/s. Comparing the packet



	10	100	1000	10000	100000
SA-Core	0.00e0%	0.00e0%	0.00e0%	9.86e0%	5.77e1%
SA-RAN	0.00e0%	0.00e0%	0.00e0%	0.00e0%	7.44e1%
NSA-Core	0.00e0%	0.00e0%	0.00e0%	0.00e0%	8.70e-2%
NSA-RAN	0.00e0%	0.00e0%	0.00e0%	0.00e0%	3.00e1%

FIGURE 10: Upload packet loss probabilities for RAN and core for different packet rates and a packet size of 128 bytes.



	128	256	512	1024	1280
SA-Core	9.86e0%	1.59e1%	2.48e0%	5.20e-4%	0.00e0%
SA-RAN	0.00e0%	0.00e0%	0.00e0%	0.00e0%	0.00e0%
NSA-Core	0.00e0%	6.20e-4%	0.00e0%	2.05e-3%	0.00e0%
NSA-RAN	0.00e0%	0.00e0%	0.00e0%	2.19e1%	3.77e1%

FIGURE 11: Upload packet loss probabilities for RAN and core for different packet sizes and a rate of 10000 packets/s.

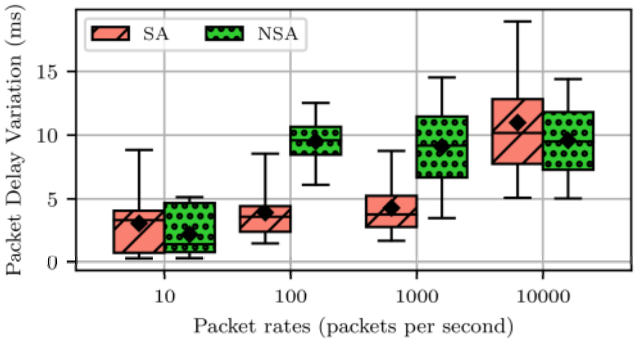


FIGURE 12: End-2-End upload Packet Delay Variation (PDV) for different packet rates; fixed packet size of 128 bytes.

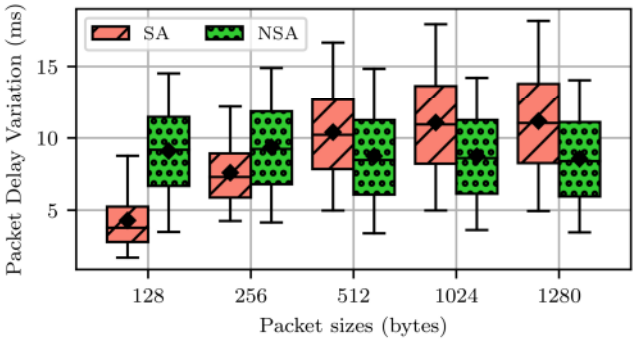


FIGURE 13: End-2-End upload Packet Delay Variation (PDV) for different packet sizes; fixed packet rate of 1000 packets/s.

loss probabilities in the SA RAN and SA core at 100000 packets/s and a packet size of 128 bytes from Figure 10 with the values in Figure 11 at a packet size of 1280 byte and 10000 packets/s, we see that, although the same bandwidth is required, there are no packet losses for the lower 10000 packets/s rate. Also, we observe from Figure 11 that for the packet sizes 128 to 512 bytes there can be significant losses in the case of the 5G SA Open5GS core, which decrease for larger packet sizes. This seemingly counterintuitive decrease of the packet loss probability for increasing packet sizes is further examined in Section IV-B, see in particular Figure 21.

For the NSA RAN, we observe about the same packet loss probabilities at 100000 packets/s and a packet size of 128 bytes in Figure 10, namely 30%, and for 1280 byte sized packets at a rate of 10000 packets/s in Figure 11 namely 37.7%. Hence, the losses in the NSA RAN are probably because the capacity is reached at about 71.7 Mbits/s and 63.8 Mbits/s, respectively. In other words, the achieved NSA upload throughput is generally independent from the packet rate and size.

3) Packet Delay Variability: PDV and IPDV

Similar to the download, there is a correlation of PDV and packet rate, as depicted in Figure 12. To improve readability we omit the 100000 packets/s rate from all following upload plots. While in NSA there is a big step-up of the PDV from 10 to 100 packets/s, in SA this PDV step-up occurs at 1000 to 10000 packets/s. In the case of SA, the PDV step-up can be partially explained with the performance issues of Open5GS in upload starting from 10000 packets/s, as examined in detail in Section IV-B. However, the specific reasons for the PDV step-up of NSA at relatively low packet rates are unknown and are an interesting direction for future research.

Packet sizes have an effect on the upload PDV for SA, as observed from Figure 13. The SA upload PDV linearly increases from 128 byte packets to 512 byte packets for SA, whereas the packet size does not influence the NSA PDV. Generally, these upload PDV values tend to be higher than

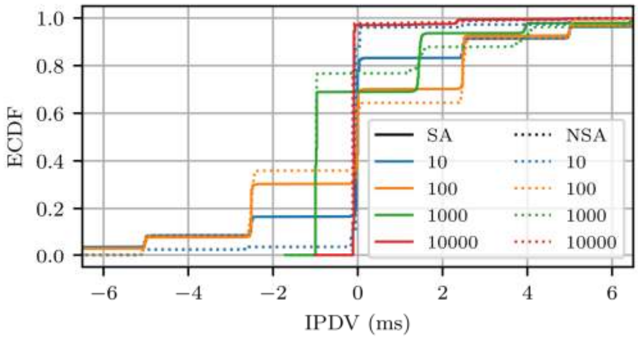


FIGURE 14: End-2-End upload Inter-Packet Delay Variation (IPDV) for different packet rates; fixed packet size of 128 bytes (corresponding download IPDV in Fig. 7).

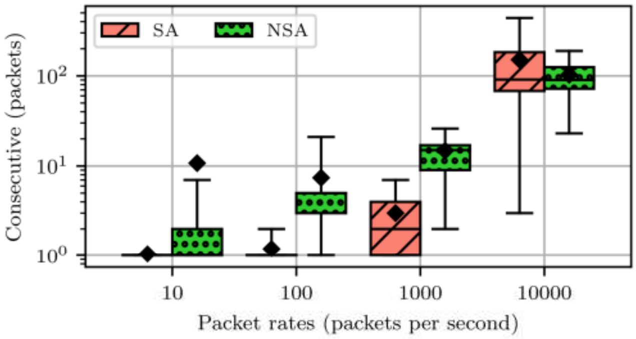


FIGURE 15: End-2-End upload downtime: Number of consecutive threshold exceeding packets for different packet rates; fixed packet size of 128 bytes.

the PDV values for download in Figure 6: for instance, for the low 10 packets/s rate, the download PDV was below 2 ms in Figure 6, but is reaching values of 5 ms and higher for upload in Figure 13. Thus, download and upload are clearly not behaving symmetrically as far as packet delay variations are concerned, which has to be accounted for by industrial control applications that require specific timing characteristics for both upload and download.

For both NSA and SA, the upload IPDV ECDF in Figure 14 has a wider spread compared to the download IPDV ECDF in Figure 7. Closer examination of the ECDF in Figure 14 reveals that the IPDV is distributed at discrete distances, which are equal for both SA and NSA. For 10 and 100 packets/s, the IPDVs are at multiples of ±2.5 ms. Starting with 1000 packets/s, the IPDVs are not symmetrically distributed around 0 ms, but start from −1 ms and then increase in 2.5 ms steps. These upload IPDV behaviors are likely due to the RAN scheduling and are distinctly different from the download IPDV in Figure 7, which had increments in ±0.5 ms steps, indicating a different internal RAN scheduling for upload vs. download. The wide spread of the upload IPDV is especially concerning for applications that require highly consistent upload packet updates, e.g., periodic measurements of a robot sensor.

4) Downtime

Also in upload, time-sensitive information must be able to be sent, so the *Downtime* metric can provide interesting insights here as well. We again set the threshold at 10 ms. In Figure 15, various packet rates are examined for a packets of size 128 bytes. For SA, the number of consecutive threshold exceeding packets increases when the packet rate is increased. However, for 10 to 1000 packets/s, about 90% of the packets have a delay less than 10 ms, as shown in Figure 9. This demonstrates again that only relying on the CDFs is not sufficient. For NSA, only 20% of delays are below 10 ms in Figure 9, except for 10 packets/s, and this corresponds to the general increased consecutive threshold-exceeding packets compared to SA in Figure 15.

To summarize, the communication from end devices to the infrastructure side, i.e., the upload, exhibits very different characteristics compared to the download. Although the highest possible throughput differs for download (1.1 Gbit/s) and upload (100 Mbit/s), the packet delay should—in theory—be the same. However, our measurements indicate a significant difference between upload delay and download delay. Hence, these measurements demonstrate that it is important to evaluate the upload delays and download delays separately.

IV. RESULTS: CORE DELAY

The motivation for utilizing 5G for Industry 4.0-related use cases is often the claimed real-time capability of 5G. However, in recent years, the core components have been implemented in software, which runs on commercial off-the-shelf (COTS) hardware and common operating systems, such as Linux. The driving factors of the core development have mainly been throughput, scalability, and cost, which can be optimized through software. However, the software-based execution on common operating systems comes at the cost of relinquishing control over computing resources, which is needed to guarantee real-time packet processing. Therefore, a close examination of the core processing delay is warranted.

A. DOWNLOAD CORE DELAY

1) Delay ECDF

This section presents the measurement results for the download core delay, which is a component of the download end-to-end delay examined in Section III-A. Figure 16 depicts the download core delay for different packet rates for packets of size 128 bytes (the corresponding end-to-end delay is shown in Figure 4). We observe from Figure 16 that for the 5G SA Open5GS core, the core delay generally tends to decrease for increasing packet rate (except for the 100000 packets/s rate, which tends to increase the core delay compared to the 10000 packets/s rate). A possible explanation for this seemingly counter-intuitive result is the batch processing by the operating system: usually multiple packets are aggregated

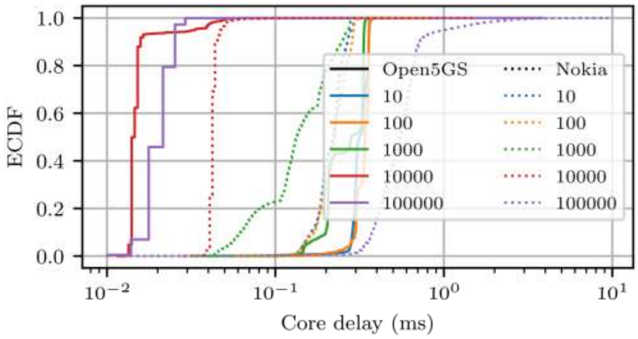


FIGURE 16: 5G SA Open5GS and 5G NSA Nokia download core delay for different packet rates; fixed 128 byte packet size.

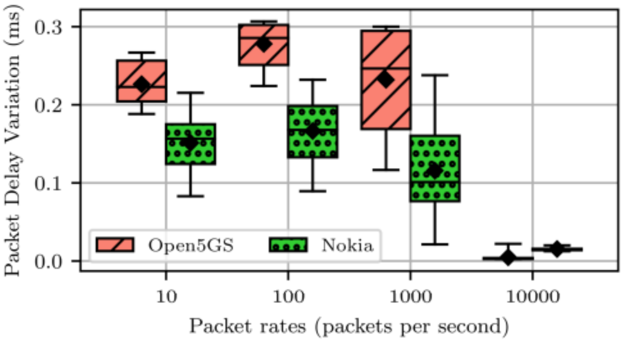


FIGURE 17: 5G SA Open5GS and 5G NSA Nokia download core download Packet Delay Variation (PDV) for different packet rates; fixed 128 byte packet size.

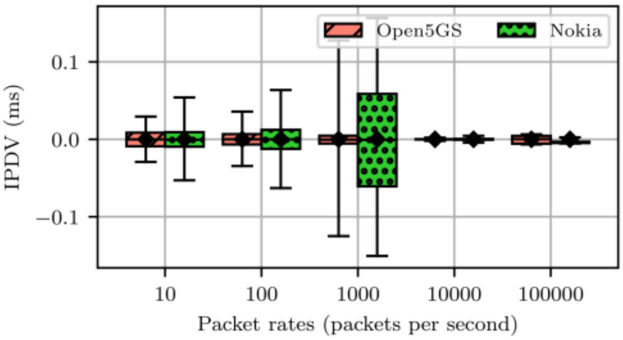


FIGURE 18: 5G SA Open5GS and 5G NSA Nokia download core Inter-Packet Delay Variation (IPDV) for different packet rates, for 128 byte packet size.

before being sent as a packet batch to an application (5G core in our case) in the user space. This packet batching increases the throughput, but also adds delay for low packet rates. However, for very high packet rates, e.g., 100000 packets per second, and small packet sizes, both cores are not able to process the packets fast enough, which results in losses, as investigated in Section III-A2.

We also observe from Figure 16 that the 5G NSA Nokia core exhibits the same general behavior as the 5G Open5GS core of decreasing core delays for increasing packet rates. Furthermore, we observe that the 5G NSA Nokia core achieves shorter core delays than the 5G SA Open5GS core for low packet rates in the 10–1000 packet/s range.

In the case of Open5GS, we observe only a slight delay increase when the packet rate increases from 10000 to 100000 packets/s. However, a more significant increase can be observed in the case of the Nokia core, namely from about 40 μs for 10000 packets/s to about 0.4–0.7 ms (for the 90% percentile) for 100000 pakets/s. (with outliers up to 10 ms, i.e., the end of the purple dotted line). Overall, at the high packet rates, the 5G NSA Nokia core appears to be slower, hence more packets queue up, resulting in higher delays (and losses, see Figure 5).

We found in additional evaluations that for both the 5G SA Open5GS core and the 5G NSA Nokia core, the core delay tends to increase only very slightly with increasing packet sizes, i.e., the packet sizes have very little effect on the core delay

2) Core Delay Variability: PDV and IPDV

Figure 17 shows the PDV for various packet rates. To improve readability the rate of 100 000 packets per second has been omitted. (Additional evaluations indicated no impact of the packet size on the PDV.) From 10 to 1000 packets/s, the Nokia core has a lower PDV compared to the Open5GS core. Both cores have the lowest PDV for a packet rate of 10000 packets per second, which corresponds to the lowest packet delays achieved in Figure 16. The decrease of PDV can be explained by the general reduction of delay at high (but not excessively high) packet rates. Overall, the PDV values of up to 0.3 ms with Open5GS and 0.25 ms with Nokia core demonstrate the need for future research and development efforts on the core delay consistency. More specifically, in a scenario with tight delay budgets, e.g., 10 ms or less, where a fixed share of the latency budget is allocated for core processing, 0.3 ms more core delay can significantly impact the overall delay budget and proper functioning of the applications. Hence, future research should not necessarily focus on minimizing the delay, but should rather focus on achieving a stable (consistent) packet delay to achieve reliable deterministic 5G network service.

Next, we consider the impact of the packet rate on the IPDV as shown in Figure 18 (the packet size was found to have no impact on the IPDV). In contrast to the PDV results, the 5G SA Open5GS core outperforms the 5G NSA Nokia core, which has a higher IPDV from 10 to 10000 packets per second. Similar to the PDV results, the IPDV decreases substantially for packet rates above 1000 packets per second for both core types. The drastic decrease of IPDV from 1000 to 10000 packets/s appears to be again due to the batch processing. Although the Nokia core has

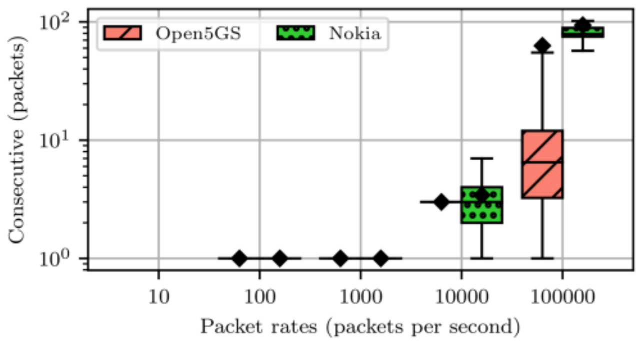


FIGURE 19: 5G SA Open5GS and 5G NSA Nokia core for download: Number of consecutive packets exceeding a 400 μs delay threshold for different packet rates; fixed 128 byte packet size.

a better PDV than Open5GS, the Nokia core has a worse IPDV, which should be examined in further detail in future research. A possible explanation could be that the Nokia core internally processes several packets at once, similar to the operating system does with batch processing. In this way, the processing of packets would be more efficient in terms of CPU utilization, throughput, and delay at low packet rates. However, between these groups of packets a delay gap would be created, resulting in a higher IPDV.

In summary, future users of 5G in automation need to closely examine the type and performance of packet processing in the core. Otherwise, undesirable behaviors could severely disrupt the automatic control of robots.

3) Downtime

For the Downtime evaluation, we set the core delay threshold to 0.4 ms, which is feasible for both cores according to Figure 16. Although this delay threshold value is arbitrary, it is a realistic assumption for future implementations of operational network that carry real-time industrial control packet traffic with an allocated delay budget.

Figure 19 depicts the downtime for different packet rates for a fixed packet size of 128 bytes. We observe from Figure 19 that for packet rates of 10 to 1000 packets/s, only up to 1 consecutive packets exceed the delay threshold. Interestingly, for 10000 packets/s, the number of consecutive packets increases for both core types despite having the lowest delay for more than 90% of the packets, as shown in Figure 16, at this packet rate. A possible explanation could be that despite batch processing at high packet rates, which decreases the delay for the majority of packets, some packets will not be included in the next batch and hence experience a relatively high delay. It is also possible that the dynamics of setting the batch sizes and forming batches [86] as well as the timing of interrupts that are triggered in the network interface processing units upon batch formation [87] contribute to the excessive delays for a number of consecutive packets. At 100000 packets per second, both cores would violate a limit of 3-6 consecutive threshold-exceeding packets.

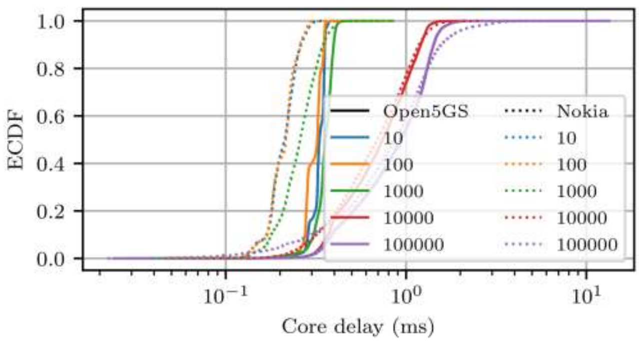


FIGURE 20: 5G SA Open5GS and 5G NSA Nokia upload core delay for different packet rates; fixed 128 byte packet size.

B. UPLOAD CORE DELAY

As we mentioned before, we already noticed performance differences in the E2E measurements during upload compared to download. To recall, in Figure 10 we measured significantly higher packet losses, especially for 5G SA Open5GS. In addition, one must keep in mind that in the upload core measurements, the specified packet rates do not necessarily correspond to the actual packet arrival rates to the core: Packet losses in the RAN can reduce the upload packet arrival rates to the core e.g., when 100000 packets/s are sent by the end device into the SA RAN, we observe from Figure 10 that only about 25600 packets/s arrive to the core.

Figure 20 shows the upload core delay for different packet rates. For low packet rates of 10 to 1000 packets the upload core delays in Figure 20 are similar to the download core delays in Figure 16. Starting with 10000 packet/s, the upload core delays for both Open5GS and Nokia in Figure 20 differ from the download core delays in Figure 16. For 10000 packets/s, 20% of the upload core delays in Figure 20 are above 1 ms and for 100000 packets/s more than 40% of the upload core delays are above 1 ms. Interestingly, this applies equally to both cores. The reasons for these pronounced core delay increases for high packet rates for upload (which did not occur for download) should be examined in follow-up research. The computing power does not seem to be the reason, since in download the end devices have to unpack the GTP packets, compared to upload where the core (with more plentiful computing resources compared to the end devices) terminates the GTP tunnel.

Furthermore, the high spread of the upload core delays in Figure 20 from about 0.4 ms to about 2 ms at high packet rates should be noted. For robotics applications, these 2 ms could be considered the worst case and budgeted accordingly. Future research should examine the use of packet processing acceleration frameworks to improve reliability.

Importantly, for different packet sizes we noticed a divergent performance compared to download, as shown in

Figure 21. For both cores, we make the counterintuitive observation that with increasing packets size, the upload core delay decreases. Whereby, the Nokia core seems to benefit more from larger packets, i.e., provides a more pronounced upload core delay decrease for the 1024 and 1280 byte packets compared to the smaller packet sizes. These decreases of the upload core delay are significant; in contrast, for download there was no significant influence of the packet size on the download core delay. This counterintuitive behavior may be due to a combination of several effects. First, the batching of packets likely plays a role, whereby the lower delays for large packets may indicate that batches may work on the basis of filling a prescribed batch size in terms of number of bytes. However, the fact that this counterintuitive behavior did not occur for the download direction indicates that additional effects are at work in the upload direction. Typically, the User Plane Function (UPF), which is responsible for packet processing, is differently implemented for upload vs. download; the differences in implementation can be verified for Open5GS core and are likely also present for the proprietary Nokia core. Additionally, it is important to keep in mind that the upload traffic arrives via the RAN to the core. Thus, the packet aggregation in the RAN scheduling and transmission cycles can lead to bursty packet traffic arrivals to the core. One advantage of our measurement setup in Figure 2 with the switch and mirror port is that our packet traffic traces that accompany this study can be analyzed for the upload data packet arrival dynamics to the core.

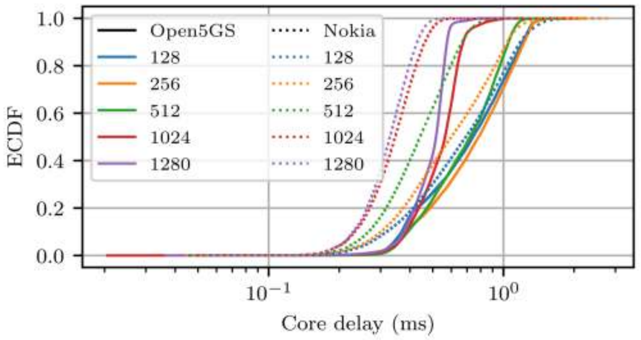


FIGURE 21: 5G SA Open5GS and 5G NSA Nokia upload core delay for different packet sizes; fixed packet rate of 10000 packets/s.

V. SUMMARY AND RESULTING RESEARCH IMPERATIVES

This section summarizes the main insights derived from the 5G NSA and SA campus network measurement results and formulates the corresponding imperatives for future research and development. We note that the summaries and research imperatives presented in this section apply generally to both 5G NSA campus networks and 5G SA campus networks. 5G NSA campus networks are commonly only viewed as an intermediate technology to bridge the gap until the 5G SA campus network technology is fully available. We have included the comparison between 5G NSA and SA campus networks in this study since some companies may want to rely on public networks, which are still NSA-based, for the foreseeable future.

A. LOW LATENCY FOR ONLY 95% OF THE PACKETS IS NOT SUFFICIENT

1) Measurement Result Summary

The ECDF plots of the download OWD in Figure 4 and the upload OWD in Figure 9 reveal that the packet delays scatter over a relatively wide range compared to the median packet delays. For instance, for SA, 95% of the download delays are in the range from 4 ms to 10 ms; whereby the delay spreads for NSA download as well as the delay spreads for upload with both SA and NSA are even wider. Robotics applications need to budget for the worst case delay. Whereby, the 10 ms may not be suitable to be considered as the worst case delay if the number of consecutive packets that exceed the 10 ms delay budget (threshold) are higher than the typically 3–6 consecutive threshold-exceeding packets that can be tolerated by robotics applications.

2) Research and Development Imperative

The wide range of packet delays observed in the measurements combined with the needs of industrial robotics applications for a prescribed delay threshold (that is exceeded only by very few consecutive packets) gives rise to a development and research imperative that focuses on the upper tail of the packet delay distribution. Specifically, future research and development should emphasize the reduction of the upper percentiles of the packet delays, e.g., the 98% percentile of the packet delay should be minimized. Furthermore, the dynamics of the packets that exceed the upper packet delay percentiles should be engineered such that only few consecutive packets exceed a given upper percentile of the delay, e.g., less than three consecutive packets should exceed the 98% packet delay percentile. This engineering of the packet delay dynamics would then permit robotics applications to budget with a given upper delay percentile with assurance that only a tolerably small number of consecutive packets exceed a tolerable delay threshold.

The shift of the research and development focus towards the reduction of the upper delay percentiles and the reduction of the number consecutive delay-threshold exceeding packets will require a fundamental shift away from the conventional packet processing paradigms that emphasize high packet throughput and short mean packet delay.

B. RETHINK THE CORE FOR INDUSTRIAL APPLICATIONS

1) Measurement Result Summary

Our core delay measurement results in Section IV demonstrate that there can be significant outliers of above one millisecond and several consecutive packets may exceed a prescribed delay threshold. This poor delay performance at the upper core delay percentiles is mainly due to the use of standard operating systems, such as Linux with standard socket programming, which are not designed for real-time packet processing.

2) Research and Development Imperative

The research agendas in the 5G and IoT topic areas have so far largely neglected the real-time packet processing in the packet core. To date, the core processing has mainly relied on conventional operating system techniques that have been designed for high throughput levels, while the latency of individual packets has typically not been specifically considered. In order to enable 5G to become suitable for industrial applications, such as industrial robot control loops with strict timing requirement, a shift in the packet core research and development agenda is needed. Instead of striving for ever higher throughput and ancillary objectives, such as low energy consumption and short mean latencies for the core packet processing, industrial applications demand research and development on operating system techniques and core packet processing techniques that prevent long packet latencies. In other words, a critical research and development imperative is to prioritize the processing of individual packets so as to reduce the upper percentiles of the packet core processing latency (at the expense of slightly reduced throughput, possibly slightly increased energy consumption, and possibly slightly increased mean core packet processing latency).

C. PACKET RATES MATTER

1) Measurement Result Summary

Our measurement results clearly indicate that the packet delays vary for different packet rates—independent of the actual consumed transmission bitrate—as shown in Figure 4 for download and Figure 9 for upload. This observed pronounced delay increase for increasing packet rates (while keeping the transmission bitrate constant) is problematic: Generally, for many industrial control applications the rate (per second) of (discrete) control signals that are delivered via typically small data packets within strict timing guarantees is critical. Oftentimes, these industrial control applications do not demand high transmission bitrates, but rather demand high rates of small packet transmissions. Also, from a practical network operations perspective, depending on the application and its desired packet rate, a different delay would have to be considered for delay budget calculations in order to account for the delay dependence on the packet rate.

2) Research and Development Imperative

5G RAN and core research and development should focus on strategies for accommodating high packet rates. The efficient transmission and processing of high packet rates needs to maintain strict timing constraints and avoid delay increases for individual packets (in order to avoid long downtimes of consecutive packets that exceed a delay threshold). This high-packet rate research will likely need to shift away from the classical batch transmission and processing strategies in order to ensure low packet transmission and processing latencies. New RAN and core processing strategies are needed that achieve fast packet transmission and processing of numerous small packets.

D. RAN SCHEDULING IS NOT YET READY FOR INDUSTRIAL APPLICATIONS

1) Measurement Result Summary

A critical aspect for industrial applications, such as robot control communication, is the guarantee of a stable delay. For the IPDV metric, we have seen that the delay variation between packets is discretely distributed in the millisecond range, as shown in Figures 7 for download and Figure 14 for upload. A delay difference of consecutive packets of multiple milliseconds, especially in upload, would disrupt many real-time robotic applications.

2) Research and Development Imperative

Real-time 5G RAN scheduling mechanisms need to be researched and developed so as to prioritize the consistency of the packet latencies. Frequent small updates of industrial control signals need to be carried with high frequency (i.e., with high packet rates), but also with high timing consistency within the packet train (sequence) for a particular control application. Novel real-time RAN scheduling mechanisms are needed to specifically consider this delay consistency requirement.

E. LOSSES ARE RARE IN THE AIR

1) Measurement Result Summary

If the wireless channel capacity is not exceeded, then losses are very rare in the wireless part (RAN), as shown in Figure 5 for download as well as Figures 10 & 11 for upload. Surprisingly, we have measured relatively high packet loss probabilities in the core, especially at small packet sizes.

2) Research and Development Imperative

Conventionally, research on reliable wireless packet transmission has focused on packet losses on the wireless channel and developed strategies to mitigate the wireless channel losses, e.g., through coding. As our measurements demonstrate, with the ongoing rise of the importance of packet compute processing in communication networks, it may be critical to research and develop strategies to mitigate packet losses across the combination of wireless channel and core processing or exclusively for the core processing.

F. UPLOAD AND DOWNLOAD DELAYS ARE NOT SYMMETRIC

1) Measurement Result Summary

Usually, when testing networks, tools such as *ping* are used that measure the RTT to obtain a rough overview of the capability of the system, such as in [58]. In contrast, our measurement setup allows for more sophisticated system evaluations by measuring the OWDs, i.e., we were able to analyze the delay for each direction (upload and download) individually. Figure 4 for download and Figure 9 for upload indicate that the delays are not symmetric.

2) Research and Development Imperative

This fairly pronounced asymmetry of the upload and download delays for both the overall one-way transmission and the core packet processing needs to be examined in further detail in future research. Industrial control systems often assume symmetrical time delays for (i) the sensor reading at the robot location to arrive at the robot control entity, and (ii) the control command issued by the control entity to arrive at the robot location. Industrial control may need to specifically account for the asymmetry of these two delays, i.e., the longer upload delay of the sensor readings compared to the shorter download delay of the control commands so as to optimize the control performance.

VI. CONCLUSION

We have developed a measurement testbed for rigorous one-way end-to-end delay measurements of the download (packet core via RAN to end device) delay as well as the upload (end device via RAN to packet core) delay in private 5G campus networks. Our measurement setup allows for isolated measurements of the RAN as well as the packet core with high precision.

Our measurement results indicate several research and development imperatives if 5G campus networks are to be utilized for industrial control applications with strict timing requirements. For instance, the upper quantiles of the packet latencies need to be reduced and the consistency of the packet delay must be improved both for the overall 5G network, as well as specifically for the packet core processing. A further important research direction is the investigation of integrating 5G campus networks into the existing IT infrastructure. For instance, future research needs to investigate the required capabilities and feasible technologies for the infrastructure that provides the connection between the RAN and the core in order to support industrial automation use cases.

The PCAPs that have been collected with our measurement setup are available as the IEEE DataPort 5G Campus Networks: Measurement Traces dataset (DOI 10.21227/xe3c-e968 [60]). In addition, the source code of the analysis tools that have been utilized to obtain the reported statistical measurement results from the PCAPs is available on GitHub [61]. These resources enable the research community to efficiently perform additional fine-grained analyses that delve deeper into individual aspects of the overall measurement study presented in this article.

REFERENCES

[1] M. Attaran, “The impact of 5G on the evolution of intelligent automation and industry digitization,” Journal of Ambient Intelligence and Humanized Computing, in print, pp. 1–17, 2021.

[2] M. Gundall, M. Strufe, H. D. Schotten, P. Rost, C. Markwart, R. Blunk, A. Neumann, J. Grießbach, M. Aleksy, and D. Wübben, “Introduction of a 5G-enabled architecture for the realization of industry 4.0 use cases,” IEEE Access, vol. 9, pp. 25 508–25 521, 2021.

[3] E. A. Oyekanlu, A. C. Smith, W. P. Thomas, G. Mulroy, D. Hitesh, M. Ramsey, D. J. Kuhn, J. D. Mcghinnis, S. C. Buonavita, N. A. Looper, M. Ng, A. Ng’oma, W. Liu, P. G. Mcbride, M. G. Shultz, C. Cerasi, and D. Sun, “A review of recent advances in automated guided vehicle technologies: Integration challenges and research areas for 5G-based smart manufacturing applications,” IEEE Access, vol. 8, pp. 202 312–202 353, 2020.

[4] A. Narayanan, A. S. D. Sena, D. Gutierrez-Rojas, D. C. Melgarejo, H. M. Hussain, M. Ullah, S. Bayhan, and P. H. J. Nardelli, “Key advances in pervasive edge computing for industrial internet of things in 5G and beyond,” IEEE Access, vol. 8, pp. 206 734–206 754, 2020.

[5] E. M. Torroglosa-Garcia, J. M. A. Calero, J. B. Bernabe, and A. Skarmeta, “Enabling roaming across heterogeneous IoT wireless networks: LoRaWAN meets 5G,” IEEE Access, vol. 8, pp. 103 164–103 180, 2020.

[6] S. Wijethilaka and M. Liyanage, “Survey on network slicing for Internet of Things realization in 5G networks,” IEEE Communications Surveys & Tutorials, vol. 23, no. 2, pp. 957–994, 2021.

[7] E. Benalia, S. Bitam, and A. Mellouk, “Data dissemination for internet of vehicle based on 5G communications: A survey,” Transactions on Emerging Telecommunications Technologies, vol. 31, no. 5, p. e3881, 2020.

[8] W. M. Chan and J. W. C. Lee, “5G connected autonomous vehicle acceptance: Mediating effect of trust in the technology acceptance model,” Asian Journal of Business Research Volume, vol. 11, no. 1, pp. 40–60, 2021.

[9] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, “A survey on 5G usage scenarios and traffic models,” IEEE Communications Surveys & Tutorials, vol. 22, no. 2, pp. 905–929, 2020.

[10] Z. Szalay, D. Ficzere, V. Tihanyi, F. Magyar, G. Soós, and P. Varga, “5G-enabled autonomous driving demonstration with a V2X scenario-in-the-loop approach,” Sensors, vol. 20, no. 24, p. 7344, 2020.

[11] S. R. Pokhrel, J. Ding, J. Park, O. S. Park, and J. Choi, “Towards enabling critical mMTC: A review of URLLC within mMTC,” IEEE Access, vol. 8, pp. 131 796–131 813, 2020.

[12] Belliardi, Rudy et al., “Use Cases IEC/IEEE 60802, V1.3,” pp. 1–74, Sep. 2018, available from http://www.ieee802.org/1/files/public/docs2018/60802-industrial-use-cases-0918-v13.pdf; Last accessed Feb. 19, 2019.

[13] M. Höyhtyä, K. Lähetkangas, J. Suomalainen, M. Hoppari, K. Kujanpää, K. Trung Ngo, T. Kippola, M. Heikkilä, H. Posti, J. Mäki, T. Savunen, A. Hulkkonen, and H. Kokkinen, “Critical communications over mobile operators’ networks: 5G use cases enabled by licensed spectrum sharing, network slicing and QoS control,” IEEE Access, vol. 6, pp. 73 572–73 582, 2018.

[14] A. Ghosh, A. Maeder, M. Baker, and D. Chandramouli, “5G evolution: A view on 5G cellular technology beyond 3GPP release 15,” IEEE Access, vol. 7, pp. 127 639–127 651, 2019.

[15] L. Nadeem, M. A. Azam, Y. Amin, M. A. Al-Ghamdi, K. K. Chai, M. F. N. Khan, and M. A. Khan, “Integration of D2D, network slicing, and MEC in 5G cellular networks: Survey and challenges,” IEEE Access, vol. 9, pp. 37 590–37 612, 2021.

[16] T. Hoeschele, C. Dietzel, D. Kopp, F. H. Fitzek, and M. Reisslein, “Importance of Internet Exchange Point (IXP) infrastructure for 5G: Estimating the impact of 5G use cases,” Telecommunications Policy, vol. 45, no. 3, p. 102091, 2021.

[17] T. W. Nowak, M. Sepczuk, Z. Kotulski, W. Niewolski, R. Artych, K. Bocianiak, T. Osko, and J.-P. Wary, “Verticals in 5G MEC-use cases and security challenges,” IEEE Access, vol. 9, pp. 87 251–87 298, 2021.

[18] A. Rostami, “Private 5G networks for vertical industries: Deployment and operation models,” in Proc. IEEE 2nd 5G World Forum (5GWF), 2019, pp. 433–439.

[19] P. Varga, J. Peto, A. Franko, D. Balla, D. Haja, F. Janky, G. Soos, D. Ficzere, M. Maliosz, and L. Toka, “5G support for industrial IoT applications–challenges, solutions, and research gaps,” Sensors, vol. 20, no. 3, p. 828, 2020.

[20] 3GPP, “Release description; Release 15 (TR 21.915),” https://portal.3gpp.org/desktopmodules/Specifications/ SpecificationDetails.aspx?specificationId=3389, [Online; accessed July 14, 2021].

[21] Bundesnetzagentur. (2019, Nov.) "Antragsverfahren für lokale 5G-Campus-Netze gestartet".

https://www.bundesnetzagentur.de/SharedDocs/Pressemitteilungen/DE/2019/20191121_lokaleFreq.html. Accessed November 11, 2020.

[22] Federal Communications Commission Report and Order, "Promoting Investment in the 3550-3700 MHz Band, FCC 18-149," https://docs.fcc.gov/public/attachments/FCC-18-149A1.pdf, Oct. 2018, [Online; accessed July 14, 2021].

[23] A. Díaz Zayas, G. Caso, Ö. Alay, P. Merino, A. Brunstrom, D. Tsolkas, and H. Koumaras, "A modular experimentation methodology for 5G deployments: The 5GENESIS approach," Sensors, vol. 20, no. 22, p. 6652, 2020.

[24] J. Kalliovaara, R. Ekman, J. Paavola, T. Jokela, J. Hallio, J. Auranen, P. Talmola, and H. Kokkinen, "Designing a testbed infrastructure for experimental validation and trialing of 5G vertical applications," in Proc. Int. Conf. on Cognitive Radio Oriented Wireless Networks, 2017, pp. 247–263.

[25] J. B. N. Koonampilli, M. Vutukuru, K. M. Sivalingam, A. Balasubramanian, R. V. Vinodh, S. Seshasayee, K. Gokhale, D. V. Kashyap, and R. R. Kamath, "Demonstration of 5G core software system in India's indigenous 5G test bed," in Proc. IEEE Int. Conf. on COMmunication Systems & NETworkS (COMSNETS), 2021, pp. 101–103.

[26] J. P. A. Mamaradlo, N. A. M. Mercado, N. J. C. Libatique, G. L. Tangonan, R. J. Solis, V. Rodriguez, B. B. Dingel, C. Pineda, and C. Lopez, "University campus 5G testbed and use case deployments in the Philippines," in Proc. SPIE Broadband Access Communication Technologies XIV, vol. 11307, 2020, pp. 1 130 704–1–1 130 704–16.

[27] M. N. Patwary, S. J. Nawaz, M. A. Rahman, S. K. Sharma, M. M. Rashid, and S. J. Barnes, "The potential short-and long-term disruptions and transformative impacts of 5G and beyond wireless networks: Lessons learnt from the development of a 5G testbed environment," IEEE Access, vol. 8, pp. 11 352–11 379, 2020.

[28] T. H. Loh, F. Heliot, D. Cheadle, and T. Fielder, "An assessment of the radio frequency electromagnetic field exposure from a massive MIMO 5G testbed," in Proc. IEEE European Conf. on Antennas and Propagation (EuCAP), 2020, pp. 1–5.

[29] A. Schumacher, R. Merz, and A. Burg, "3.5 GHz coverage assessment with a 5G testbed," in Proc. IEEE 89th Vehicular Technology Conf. (VTC2019-Spring), 2019, pp. 1–6.

[30] M. Bärring, O. Iupikov, A. A. Glazunov, M. Ivashina, J. Berglund, B. Johansson, J. Stahre, F. Harrysson, U. Engström, and M. Friis, "Factory radio design of a 5G network in offline mode," IEEE Access, vol. 9, pp. 23 095–23 109, 2021.

[31] I. Freire, C. Novaes, I. Almeida, E. Medeiros, M. Berg, and A. Klautau, "Clock synchronization algorithms over PTP-unaware networks: Reproducible comparison using an FPGA testbed," IEEE Access, vol. 9, pp. 20 575–20 601, 2021.

[32] I. Freire, I. Almeida, E. Medeiros, M. Berg, C. Lu, E. Trojer, and A. Klautau, "Testbed evaluation of distributed radio timing alignment over Ethernet fronthaul networks," IEEE Access, vol. 8, pp. 87 960–87 977, 2020.

[33] M. Irazabal, E. Lopez-Aguilera, I. Demirkol, R. Schmidt, and N. Nikaein, "Preventing RLC buffer sojourn delays in 5G," IEEE Access, vol. 9, pp. 39 466–39 488, 2021.

[34] T. K. Le, U. Salim, and F. Kaltenberger, "An overview of physical layer design for ultra-reliable low-latency communications in 3GPP releases 15, 16, and 17," IEEE Access, vol. 9, pp. 433–444, 2021.

[35] T. Magounaki, F. Kaltenberger, and R. Knopp, "Modeling the distributed MU-MIMO OAI 5G testbed and group-based OTA calibration performance evaluation," in Proc. IEEE Int. Works. Sig. Proc. Adv. in Wirel. Commun. (SPAWC), 2020, pp. 1–5.

[36] D. Marabissi, L. Mucchi, S. Caputo, F. Nizzi, T. Pecorella, R. Fantacci, T. Nawaz, M. Seminara, and J. Catani, "Experimental measurements of a joint 5G-VLC communication for future vehicular networks," Journal of Sensor and Actuator Networks, vol. 9, no. 3, p. 32, 2020.

[37] K. Zheng, D. Wang, Y. Han, X. Zhao, and D. Wang, "Performance and measurement analysis of a commercial 5G millimeter-wave network," IEEE Access, vol. 8, pp. 163 996–164 011, 2020.

[38] A. Filali, A. Abouaomar, S. Cherkaoui, A. Kobbane, and M. Guizani, "Multi-access edge computing: A survey," IEEE Access, vol. 8, pp. 197 017–197 046, 2020.

[39] P. Shantharama, A. S. Thyagaturu, N. Karakoc, L. Ferrari, M. Reisslein, and A. Scaglione, "LayBack: SDN management of multi-access edge computing (MEC) for network access services and radio resource sharing," IEEE Access, vol. 6, pp. 57 545–57 561, 2018.

[40] F. Spinelli and V. Mancuso, "Toward enabled industrial verticals in 5G: A survey on MEC-based approaches to provisioning and flexibility," IEEE Communications Surveys & Tutorials, vol. 23, no. 1, pp. 596–630, 2020.

[41] G. Akilandeswary and M. L. Manickam J., "Next generation network coding technique for IoT," in Proc. IEEE 11th Int. Conf. on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1–6.

[42] A. Cohen, G. Thiran, V. B. Bracha, and M. Medard, "Adaptive causal network coding with feedback for multipath multi-hop communications," IEEE Transactions on Communications, vol. 69, no. 2, pp. 766–785, 2021.

[43] V. Nguyen, E. Tasdemir, G. T. Nguyen, D. E. Lucani, F. H. Fitzek, and M. Reisslein, "DSEP Fulcrum: Dynamic sparsity and expansion packets for Fulcrum network coding," IEEE Access, vol. 8, pp. 78 293–78 314, 2020.

[44] T. Dreibholz, "Flexible 4G/5G testbed setup for mobile edge computing using OpenAirInterface and Open Source MANO," in Proc. Workshops Int. Conf. Adv. Inform. Netw. Appl. Springer, Cham, Switzerland, 2020, pp. 1143–1153.

[45] A. Esmaeily, K. Kralevska, and D. Gligoroski, "A cloud-based SDN/NFV testbed for end-to-end network slicing in 4G/5G," in Proc. 6th IEEE Conf. on Network Softwarization (NetSoft), 2020, pp. 29–35.

[46] A. A. Kherani, G. Shukla, S. Sanadhya, N. Vasudev, M. Ahmed, A. S. Patel, R. Mehrotra, B. Lall, H. Saran, M. Vutukuru et al., "Development of MEC system for indigenous 5G test-bed," in Proc. IEEE Int. Conf. on COMmunication Systems & NETworkS (COMSNETS), 2021, pp. 131–133.

[47] G. Nardini, G. Stea, A. Virdis, D. Sabella, and P. Thakkar, "Using Simu5G as a realtime network emulator to test MEC apps in an end-to-end 5G testbed," in Proc. IEEE Int. Symp. on Personal, Indoor and Mobile Radio Commun. (PIMRC), 2020, pp. 1–7.

[48] 5GTNF. (2020, Sep.) "5GTNF Project Website". https://5gtnf.fi/. Accessed September 24, 2020.

[49] Telefonaktiebolaget LM Ericsson. (2020, Sep.) Europe's largest 5G research network goes live in Germany. https://www.ericsson.com/en/news/2020/5/europes-largest-5g-research-network-goes-live. Accessed September 24, 2020.

[50] Fraunhofer HHI. (2020, Sep.) "Fraunhofer HHI und 5G Berlin nehmen 5G-Testfeld zur Erprobung neuer 5G-Technologien in Berlin-Charlottenburg in Betrieb". https://newsletter.fraunhofer.de/-viewonline2/17386/517/9/14SHcBTt/CeaHnvIdmd/1. Accessed September 25, 2020.

[51] Fraunhofer FOKUS. (2020, Sep.) "NGNI 5G Playground". https://www.fokus.fraunhofer.de/go/en/fokus_testbeds/5g_playground. Accessed September 24, 2020.

[52] M. Ghassemian, P. Muschamp, and D. Warren, "Experience Building A 5G Testbed Platform," 2020. [Online]. Available: https://arxiv.org/abs/2008.01628

[53] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Open, programmable, and virtualized 5G networks: State-of-the-art and the road ahead," Computer Networks, vol. 182, p. 107516, 2020.

[54] A. Panicker, O. Ozdemir, M. L. Sichitiu, I. Guvenc, R. Dutta, V. Marojevic, and B. Floyd, "AERPAW emulation overview and preliminary performance evaluation," Computer Networks, vol. 194, p. 108083, 2021.

[55] D. Raychaudhuri, I. Seskar, G. Zussman, T. Korakis, D. Kilper, T. Chen, J. Kolodziejski, M. Sherman, Z. Kostic, X. Gu et al., "Challenge: COSMOS: A city-scale programmable testbed for experimentation with advanced wireless," in Proceedings ACM 26th Annual Int. Conf. on Mobile Computing and Networking (MobiCom), 2020, pp. 1–13.

[56] J. Breen, A. Buffmire, J. Duerig, K. Dutt, E. Eide, M. Hibler, D. Johnson, S. K. Kasera, E. Lewis, D. Maas et al., "POWDER: Platform for open wireless data-driven experimental research," in Proc. ACM 14th Int. Workshop on Wireless Network Testbeds, Experimental evaluation & Characterization, 2020, pp. 17–24.

[57] M. Laner, P. Svoboda, P. Romirer-Maierhofer, N. Nikaein, F. Ricciato, and M. Rupp, "A comparison between one-way delays in operating HSPA and LTE networks," in Proc. IEEE Int. Symp. on Modeling and Optim. in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2012, pp. 286–292.

[58] D. Xu, A. Zhou, X. Zhang, G. Wang, X. Liu, C. An, Y. Shi, L. Liu, and H. Ma, "Understanding operational 5G: A first measurement study on its coverage, performance and energy consumption," in Proc. ACM SigComm, 2020, pp. 479–494.

[59] G. Soós, D. Ficzere, P. Varga, and Z. Szalay, "Practical 5G KPI measurement results on a non-standalone architecture," in Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS), 2020, pp. 1–5.

[60] J. Rischke, “5G Campus Networks: Measurement Traces,” 2021. [Online]. Available: https://dx.doi.org/10.21227/xe3c-e968

[61] ——, “5G Campus Networks: A First Measurement Study (Source Code),” 2021. [Online]. Available: https://github.com/justus-comnets/5g-campus-measurements

[62] S. Lee, “Open5GS,” https://open5gs.org/, [Online; accessed July 14, 2021].

[63] P. Emmerich, S. Gallenmüller, D. Raumer, F. Wohlfart, and G. Carle, “MoonGen: A scriptable high-speed packet generator,” in Proc. ACM Internet Measurement Conf. (IMC), 2015, pp. 275–287.

[64] Intel Corporation, “Intel ethernet controller x550,” Datasheet Rev. 2.6, January 2021.

[65] P. Emmerich, S. Gallenmüller, G. Antichi, A. W. Moore, and G. Carle, “Mind the gap – a comparison of software packet generators,” in Proc. ACM/IEEE Symp. on Architectures for Netw. and Commun. Sys. (ANCS), 2017, pp. 191–203.

[66] COMMSCOPE, “Ruckus ICX 7850 Switch,” Datasheet https://www.commscope.com/globalassets/digizuite/61734-ds-icx-7850.pdf, [Online; accessed July 14, 2021].

[67] D. Cerović, V. Del Piccolo, A. Amamou, K. Haddadou, and G. Pujolle, “Fast packet processing: A survey,” IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 3645–3676, 2018.

[68] P. Emmerich, D. Raumer, S. Gallenmüller, F. Wohlfart, and G. Carle, “Throughput and latency of virtual switching with Open vSwitch: A quantitative analysis,” Journal of Network and Systems Management, vol. 26, no. 2, pp. 314–338, 2018.

[69] C. Stylianopoulos, M. Almgren, O. Landsiedel, M. Papatriantafilou, T. Neish, L. Gillander, B. Johansson, and S. Bonnier, “On the performance of commodity hardware for low latency and low jitter packet processing,” in Proc. ACM Int. Conf. on Distributed and Event-Based Systems, 2020, pp. 177–182.

[70] Z. Xiang, F. Gabriel, E. Urbano, G. T. Nguyen, M. Reisslein, and F. H. P. Fitzek, “Reducing latency in virtual machines: Enabling tactile internet for human-machine co-working,” IEEE Journal on Selected Areas in Communications, vol. 37, no. 5, pp. 1098–1116, 2019.

[71] P. Emmerich, D. Raumer, F. Wohlfart, and G. Carle, “A study of network stack latency for game servers,” in Proc. IEEE 13th Annual Workshop on Network and Systems Support for Games, 2014, pp. 1–6.

[72] A. Morton and B. Claise, “Packet delay variation applicability statement,” Internet Requests for Comments, RFC Editor, RFC 5481, March 2009.

[73] M. S. Brunella, G. Belocchi, M. Bonola, S. Pontarelli, G. Siracusano, G. Bianchi, A. Cammarano, A. Palumbo, L. Petrucci, and R. Bifulco, “hXDP: Efficient software packet processing on FPGA NICs,” in Proc. 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2020, pp. 973–990.

[74] S. Lange, A. Nguyen-Ngoc, S. Gebert, T. Zinner, M. Jarschel, A. Köpsel, M. Sune, D. Raumer, S. Gallenmüller, G. Carle, and P. Tran-Gia, “Performance benchmarking of a software-based LTE SGW,” in Proc. 11th Int. Conf. on Network and Service Management (CNSM), 2015, pp. 378–383.

[75] L. Linguaglossa, S. Lange, S. Pontarelli, G. Rétvári, D. Rossi, T. Zinner, R. Bifulco, M. Jarschel, and G. Bianchi, “Survey of performance acceleration techniques for network function virtualization,” Proc. IEEE, vol. 107, no. 4, pp. 746–764, 2019.

[76] L. Linguaglossa, D. Rossi, S. Pontarelli, D. Barach, D. Marjon, and P. Pfister, “High-speed data plane and network functions virtualization by vectorizing packet processing,” Computer Networks, vol. 149, pp. 187–199, 2019.

[77] P. Shantharama, A. S. Thyagaturu, and M. Reisslein, “Hardware-accelerated platforms and infrastructures for network functions: A survey of enabling technologies and research studies,” IEEE Access, vol. 8, pp. 132 021–132 085, 2020.

[78] M. A. Vieira, M. S. Castanho, R. D. Pacífico, E. R. Santos, E. P. C. Júnior, and L. F. Vieira, “Fast packet processing with eBPF and XDP: Concepts, code, challenges, and applications,” ACM Computing Surveys (CSUR), vol. 53, no. 1, pp. 1–36, 2020.

[79] Z. Li, M. A. Uusitalo, H. Shariatmadari, and B. Singh, “5G URLLC: Design challenges and system concepts,” in Proc. 15th Int. Symp. on Wireless Communication Systems (ISWCS), 2018, pp. 1–6.

[80] D. Cavalcanti, S. Bush, M. Illouz, G. Kronauer, A. Regev, and G. Venkatesan, “Wireless TSN-definitions use cases & standards roadmap,” AVNU Alliance, pp. 1–16, 2020.

[81] C. Fischer, D. Krummacker, M. Karrenbauer, and H. D. Schotten, “A modular design concept for shaping future wireless TSN solutions,” Information, vol. 12, no. 1, p. 12, 2021.

[82] M. Gundall, C. Huber, P. Rost, R. Halfmann, and H. D. Schotten, “Integration of 5G with TSN as prerequisite for a highly flexible future industrial automation: Time synchronization based on IEEE 802.1 AS,” in IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society. IEEE, 2020, pp. 3823–3830.

[83] J. Ohms, M. Böhm, and D. Wermser, “Concept of a TSN to real-time wireless gateway in the context of 5G URLLC,” in Proc. IEEE Int. Conf. on Wireless Networks and Mobile Commun. (WINCOM), 2020, pp. 1–6.

[84] S. Bhattacharjee, K. Katsalis, O. Arouk, R. Schmidt, T. Wang, X. An, T. Bauschert, and N. Nikaein, “Network slicing for TSN-based transport networks,” IEEE Access, vol. 9, pp. 62 788–62 809, 2021.

[85] A. M. Romanov, F. Gringoli, and A. Sikora, “A precise synchronization method for future wireless TSN networks,” IEEE Transactions on Industrial Informatics, vol. 17, no. 5, pp. 3682–3692, 2021.

[86] P. Okelmann, L. Linguaglossa, F. Geyer, P. Emmerich, and G. Carle, “Adaptive batching for fast packet processing in software routers using machine learning,” in Proc. 7th IEEE Int. Conf. on Network Softwarization (NetSoft),, 2021.

[87] P. Emmerich, D. Raumer, A. Beifuß, L. Erlacher, F. Wohlfart, T. M. Runge, S. Gallenmüller, and G. Carle, “Optimizing latency and CPU load in packet processing systems,” in Proc. Int. Symp. on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2015, pp. 1–8.