Lisa Wernet, Fridolin Siegmund, Ralf Kundel, Tobias Meuser, Ralf Steinmetz. *Fault-tolerance in Future RAN: Enabling Redundancy for Efficient Recovery of RAN Network Functions.* To appear in the Proceedings of IEEE Conference on Network Function Virtualization and Software Defined Networks 2023, IEEE, 2023.

Fault-tolerance in Future RAN: Enabling Redundancy for Efficient Recovery of RAN Network Functions

Lisa Wernet*, Fridolin Siegmund*, Ralf Kundel, Tobias Meuser, Ralf Steinmetz
Multimedia Communications Lab, Technical University of Darmstadt
{firstname.lastname}@kom.tu-darmstadt.de

***Abstract*—In recent years, the reliability and resilience of communication networks and cellular networks in particular has become of increasing importance. For cellular networks, disaggregation and virtualization of RAN network functions offer the opportunity to improve the overall resilience of future networks. That is, improving the resilience of individual network functions is much less complex compared to improving the resilience of the overall communication network. Overall, the resilience of individual network functions increases resilience of the communication network.**

In this paper, we propose a mechanism to enable redundant network functions in the Radio Access Network (RAN) to reduce its total downtime in case of disruptions or failures. If a network function is disrupted, the redundant network function in warm standby is activated to recover the functionality of the disrupted network function. Different Radio Intelligent Controller (RIC) services are used to monitor RAN components and control the efficient recovery of disrupted network functions. We show that the proposed mechanism significantly reduces the overall downtime of the RAN compared to restarting a disrupted network function. Our prototypical implementation on the example of the Distributed Unit (DU) for replacing a disrupted network function in warm standby reduces the downtime by 76.86% (median 41.13%) compared to restarting the disrupted network function.

***Index Terms*—6G, RIC, Resilience, O-RAN, Redundancy, VNF, Standby Management**

I. INTRODUCTION

Evolving technologies like VR applications, smart factories, or vehicular communication systems are designed with the promise that future cellular networks will provide the necessary performance and reliable access to those technologies [1]. With rising dependencies from wireless access networks, those networks must address resiliency as a key functionality besides performance optimization for low-latency and high throughput. Release 16 of the 5G network specification already included measures for high-reliability, i.e., several proposals for duplicating the user plane between a monolithic Radio Access Network (RAN) and the 5G core network [2]. However, many details about the different proposals still need to be considered. Additionally, those measures for traffic duplication are resource-intensive, inflexible and may induce other difficulties such as dropping the redundant traffic flows.

The disaggregation and virtualization of RANs offer the opportunity to assess the impact of failures in small functional units, to react to failures individually, and to deploy mechanisms that increase the self-healing capacities of the overall system. In this paper, we propose a mechanism to enable highly resilient network functions in the RAN through redundant network functions. Due to their statelessness, redundant network functions are commonly considered easy to migrate. However, network functions employed in the RAN are heavily state dependent, drastically increasing the complexity of a successful migration. As a first step for this migration, we provision redundant network functions in warm standby in the network. Once the disruption of a network function is detected, the redundant network function is activated and replaces the disrupted network function. Even though the state cannot be fully transferred between the failed and the replacing network function due to the high Quality of Service (QoS) requirements for the synchronization, we can show that the recovery of network functions can be significantly improved in an experimental implementation using OpenAirInterface (OAI).

Our contributions are as follows:

- A monitoring mechanism for the surveillance of the O-RAN network functions.
- A recovery strategy based on warm standby of network functions.
- A prototypical implementation for the replacement of distributed units in the O-RAN architecture.

The remainder of this paper is structured as follows: Section II gives an overview of related publications. In Section III, we introduce the system architecture and propose our novel approach for the recovery of network functions. Section IV focuses on the practical implementation of our approach, and in Section V, we prove that we can significantly reduce the downtime of the RAN. Finally, in Section VI, we conclude our work and offer insights about future work.

II. BACKGROUND AND RELATED WORK

In cellular communication networks, the Radio Access Network (RAN) acts as the interface between the User Equipment (UE) and the Core Network. It transfers data from the packet-based wired network through an air interface to UEs and vice

---

*Authors contributed equally to the paper.

Lisa Wernet, Fridolin Siegmund, Ralf Kundel, Tobias Meuser, Ralf Steinmetz. *Fault-tolerance in Future RAN: Enabling Redundancy for Efficient Recovery of RAN Network Functions.* To appear in the Proceedings of IEEE Conference on Network Function Virtualization and Software Defined Networks 2023, IEEE, 2023.

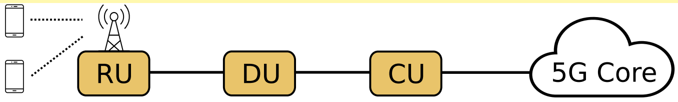


Fig. 1: Simplified O-RAN architecture with Virtual Network Functions (VNFs) and RU hardware connected to the 5G core.

versa. While classic approaches have built up the RAN monolithically, in the O-RAN approach as shown in Figure 1, the RAN consists of disaggregated Network Functions (NFs) such as the Central Unit (CU), Distributed Unit (DU) and Radio Unit (RU), which each take over corresponding RAN layer functionalities [3], [4]. The different layers of the RAN are distributed among the O-RAN components (CU, DU and RU), while the exact distribution depends on the implemented split option [5]. While the functionality of CU and DU is decoupled from hardware and can be entirely realized in software, the RU is required to transmit and receive the actual radio waves to and from UEs and therefore is built with specialized hardware and unified interface and protocols to the DU [6]. Hence, the disaggregation and specified interfaces allow the mixing of different vendors of the RU and implementations of DU and CU. The network between RU and DU is typically called fronthaul, while the network between DU and CU is called midhaul. The midhaul consists of the F1 interface. It is split up into the F1-C interface for control plane traffic and the F1-U interface for user plane traffic. Control plane messages are exchanged using the F1 Application Protocol (F1AP) with Stream Control Transmission Protocol (SCTP) as transport protocol. User plane data between DU and CU is exchanged via the General Packet Radio Service Tunneling Protocol for User Plane (GTP-U) with User Datagram Protocol (UDP) as transport protocol. The network between CU and the 5G core is named backhaul, terminating either in a software- or hardware-based network function [7].

One key concept this work relies on is handling a failover mechanism between redundant RAN VNFs. Different types of standby for redundant VNFs can be delimited, including hot, warm, and cold standby [8]. While in hot standby mode, a VNF is required to be synchronized permanently with its counterpart in operation, cold standby refers to a VNF in an utterly inactive state [9]. In between these two standby types, a VNF may be in warm standby mode, i.e., it is active but operates with fewer tasks than a VNF in hot standby, and the state is not entirely synced with its counterpart.

In addition to the disaggregation of the RAN, the O-RAN Alliance defines the Radio Intelligent Controller (RIC) [10]. The RIC is connected to VNFs in the RAN, i.e., CU and DU instances, over different interfaces. The architecture of the RIC comprises the near-real time RIC functions and a Service Management and Orchestration (SMO) framework containing the non-real time RIC functions. In this work, we realize the functionality of such a RIC for managing the recovery of VNFs. Schmidt et al. introduced FlexRIC, a Software Development Kit (SDK) to build near-real time RAN controllers [11]. FlexRIC is now an active project from OpenAirInterface (OAI) [12]. FlexRIC implements a near-real time RIC and uses E2 interfaces towards the RAN that are compliant with the interfaces defined by the O-RAN Alliance. Standardized service models can be used to monitor RAN VNFs. Monitoring data can be accessed and processed via dedicated xApps. Those xApps can then control certain functionalities of the RAN. The E2 interface is used to induce and configure policies in the RAN. Policies influence specific existing RAN functions for the following use cases: traffic steering, QoS based resource optimization, RAN Slice SLA Assurance, Massive MIMO Optimization, QoE Optimization [10]. So, FlexRIC would allow accessing states from multiple 5G layers, e.g., for failure detection or the implementation of state synchronization in hot standby scenarios. However, it is not possible to manage the operating state and influence standby modes of NFs with FlexRIC. In the current RIC design of the O-RAN Alliance, heartbeat messages from DUs and CUs are sent over the O1 interface to the SMO framework [3]. This interface is generally used for various tasks related to fault, configuration, accounting, performance, security (FCAPS). For our work, we need to implement the SMO related tasks in the near-real time RIC to be able to realize the efficient recovery of faulty VNFs.

Ramanathan et al. presented their approach to migrate containerized RAN applications between different hosts, especially the CU [13]. The authors distinguish between two migration operations: First is the stateless migration, where a fresh containerized CU instance is created at the destination host, and the old one is destroyed. The states are lost during the migration as the container is restarted. Second is the stateful migration, where a whole containerized CU is migrated to the destination host. Here, the container's operating state is frozen, preserving all states. At the destination host, the operation is resumed from the frozen state. They modified the underlying SCTP implementation in the Linux kernel to prevent a shutdown of the SCTP connection during migration of the RAN container [13]. The user plane communication is based on UDP as transport protocol which is not stateful in the network stack. Without warm standby, other research has shown that RAN VNFs require multiple seconds up to minutes to be operational [13]. Ramanathan et al. observe a service downtime at the UE of $12\,s$ for stateless migration and $4\,s$ for stateful migration. The authors conclude that improved state migration mechanisms would reduce the UE service downtime. Ramanathan's approach is limited by realizing the VNFs in the RAN as whole VM instances coming with negligible capabilities of managing the VNFs internally [13]. Also, the authors realized the migration only with a CU, for DU the evaluation may be different or not possible due to timing constraints in the lower 5G layers processed by DUs. A smaller ping interval could improve the time resolution of the evaluation.

Leiter et al. implemented failover procedures for the User Plane Function (UPF) in the 5G core by instantiating a new UPF container after the previous one is disrupted [14]. The authors explain that they consider the failover a success based

Lisa Wernet, Fridolin Siegmund, Ralf Kundel, Tobias Meuser, Ralf Steinmetz. *Fault-tolerance in Future RAN: Enabling Redundancy for Efficient Recovery of RAN Network Functions.* To appear in the Proceedings of IEEE Conference on Network Function Virtualization and Software Defined Networks 2023, IEEE, 2023.

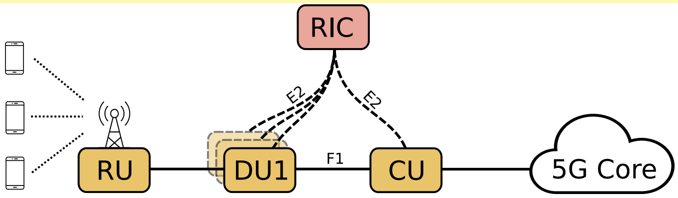


Fig. 2: The active Radio Access Network (RAN) consists of a RU and the VNFs DU1 and CU, connected via the F1 interface. Additional DU instances are waiting in warm standby. The Radio Intelligent Controller (RIC) service communicates with all available DU and CU instances over the E2 interface.

on the ready state and not on the communication state, e.g., if the functionality of the UPF recovered or if startup procedures are required. They measure a minimum of 24.97 s and a maximum of 82.05 s for the failover time. For a production environment, the researchers suggest a VNF in hot redundancy mode, which is already started before the disruption occurs, to reduce failover time.

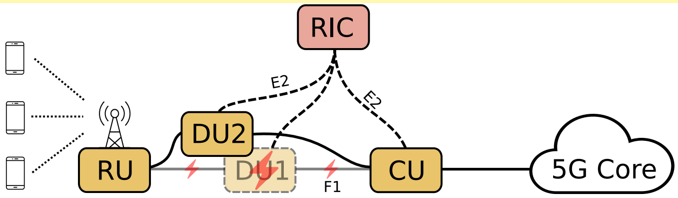
Although there are standards and implementations for managing RANs VNFs in an operational state, failover mechanisms for individual NFs still need to be considered. Additionally, the current standardization of the RIC and protocols may need to be revised for management of VNFs in standby as well as real-time use cases [9].

III. DESIGN

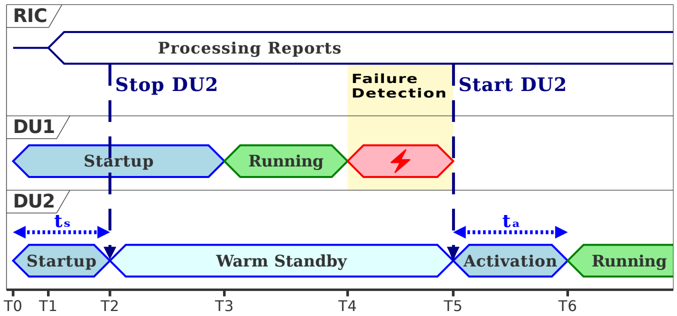
This work aims at an efficient and feasible detection and mitigation of failures of independent Virtual Network Functions (VNFs). We consider an Open Radio Access Network (O-RAN) with a fronthaul split between Radio Unit (RU) and Distributed Unit (DU) (option 7.2) and a midhaul split between DU and Central Unit (CU) (option 2) as shown in Figure 2. As DUs and CUs are virtualized network functions, other inactive instances beyond the currently active instance may exist in warm standby. These inactive instances are potential candidates for the replacement of the active instance in case of failure. For that purpose, we assume that the RIC provides the following functionalities:

- Surveillance of availability of O-RAN VNFs.
- Monitoring selected states of O-RAN VNFs to be processed by xApps for automatic failure detection.
- Flexible adaptation of monitoring.
- Control of the activity levels of redundant components.

For our use case, the detection and mitigation of faults, we need to monitor whether RAN components are available and working as intended. While heartbeats are an indicator for the availability of a RAN component, messages containing state values from RAN components can be used to monitor and verify the functionality of internal procedures of a component. In our design, we utilize the E2 `REPORT` messages to accomplish this task, which allows us to collect fine-grained data of the component. For the control of RAN components, we need to modify the E2 interface so that we can influence the activity of a component (i.e., switch from warm standby to active) with a `POLICY` message rather than separate existing functions of a component.



(a) Topology of the failure scenario with faulty DU1 being replaced by DU2.



(b) Sequence of event: normal operation, failure of DU2 and recovery to normal operation.

Fig. 3: Recovery of RAN functionality by using redundant DU instances.

During normal operation, CU and DU instances will periodically report their internal states to the RIC. The RIC assesses these reports to track the availability as well as the correct functioning of all VNFs. It modulates the frequency of those `REPORT` messages utilizing `POLICY` messages to RAN VNFs. To manage several redundant DU instances - all but one instance are in warm standby mode - several `POLICY` messages are used to tune the activity level of individual instances. The controller uses a `POLICY` message to steer if an instance starts as an active instance or waits in standby mode until it receives another `POLICY` message, which triggers its activation. Any DU in warm standby is already running, but connections to other O-RAN components than the RIC still need to be established. Also, there needs to be a way to stop and terminate faulty active DU instance as otherwise a replacement may not be possible. If the RIC recognizes a failure in the active DU, it forces this instance to be stopped and activates a redundant instance.

Figure 3 shows a scenario where the active instance DU1 failed and was replaced by the redundant instance DU2.

Figure 3a depicts the topology and Figure 3b visualizes the chronological order of events of this scenario. For better readability, this figure only depicts DU instances and the RIC while CU and RU are omitted. In the beginning, at *T0*, while all RAN components are starting, the RIC is already running. At *T1*, RAN VNFs are connected to the RIC over E2, but the 5G functions are not yet ready. In the next step, at *T2*, the controller interrupts the startup process of the redundant DU2 to place the redundant network function (DU2) available in warm standby. At *T3*, all other RAN components, including DU1, have finished their startup process, and the RAN starts its normal operations within the 5G network. The controller tracks the correct behavior of all RAN components by processing received `REPORT` messages. At *T4*, DU1 is disrupted and stops transmitting `REPORT` messages to the RIC. The RIC waits for new messages from DU1 until a timeout is triggered. Then, at *T5*, the controller activates DU2 by sending a `POLICY` message to this inactive instance. Additionally, the controller enforces the proper termination of the faulty DU1 to make sure that this instance does not interfere in the activation of the redundant instance, e.g., by blocking the establishment of the F1 connection between DU2 and CU. At *T6*, DU2 finishes its remaining startup procedures, and the RAN resumes its normal operation. The length for the activation of DU2, t_a is significantly shorter than the total startup time of DU1. The time saving between the total startup time of DU1 and the activation time of DU2, t_a, is about the size of t_s. Splitting up the startup procedure of redundant DU instances efficiently reduces the time for replacing faulty instances. This time saving reduces the overall downtime of the RAN.

With our approach, the downtime of network functions in the RAN can be significantly reduced depending on the implementation. The downtime reduction depends, beyond others, on the duration of the activation of a network function in warm standby. The whole mitigation strategy is coordinated by the RIC, a central component in the RAN. The RIC detects the failure of a network function and coordinates the mitigation strategy subsequently. As the RIC itself is a purely software-based component, it can also be run in a distributed manner to alleviate disruptions of the RIC itself. In the next section, we show how our design can be implemented in the open-source RAN implementation of OpenAirInterface (OAI).

IV. IMPLEMENTATION

The RAN implementation in this work is based on the open-source implementation OAI. At the time of this work, the OAI 5G RAN implementation offers only a higher layer split (option 2) between CU and an aggregated RU/DU component. Nevertheless, the functionalities of both components are divided from each other. Tasks that belong to either RU or DU run in separate threads. Further, the RU part of the aggregated component is stateless. Therefore, as our implementation changes states solely in the DU, it is generally applicable for a fully disaggregated O-RAN with an additional fronthaul split between RU and DU as described in Section III.

In our experimental testbed, we use the remote procedure call framework gRPC to implement messaging procedures over the E2 interface between RIC and RAN components [15]. Figure 4 shows the connection between RAN components and RIC. There are two different message types, i.e., unary and stream messages. `REPORT` messages are implemented as unary messages. VNFs use these `REPORT` messages to periodically send data to the controller, while stream messages are used to transmit `POLICY` messages from the controller to VNFs. In the following, we describe the extension of the OAI RAN components that are necessary to support the RIC services.

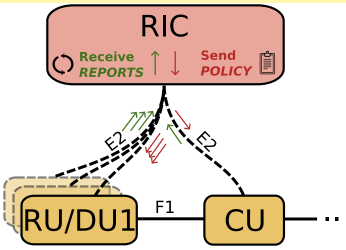


Fig. 4: Communication between RIC and RAN components (CUs and DUs).

A. Adaptive Monitoring and Surveillance

Our RU/DU and CU instances can periodically transmit state values to our RIC, for example, connected User Equipments (UEs), Channel Quality Indicator (CQI) and further transceiving statistics. Also, the local VNF configuration, such as configured network interfaces and node type, is sent to the RIC. The frequency of those `REPORT` messages can be set by `POLICY` messages from the controller.

B. Control of Redundant VNFs

To enable a warm standby of RU/DU instances, we equip all instances with halting points at the corresponding locations that handle the establishment of the radio connection to UEs and of the Stream Control Transmission Protocol (SCTP) connection to the CU. The RIC can activate those halting points by sending a `policy` message to the respective instance. For this reason, the connection to the RIC is set up before the RU/DU instance finishes its startup procedure and the RIC can activate the halting points of the RAN functions. If the halting points are active during the startup of RAN functions, the RU/DU instance waits in warm standby until a second `POLICY` message finally triggers the establishment of the radio link and the F1 interface to the CU. If these halting points stay inactive during startup, the RU/DU instance will directly act as active instance, i.e., connecting to the CU and turning on its radio. To make this mechanism possible, we extended the CU with the capability to handle the disruption of the link to the first RU/DU instance and the connection establishment to the new RU/DU instance. Instead of always using the static IP address from the configuration file, our CU implementation supports dynamic reconfiguration of the DU F1 endpoint IP address by learning the address of a new RU/DU instance directly from the SCTP handshake. With that, our implementation supports the replacement of DUs during runtime, which drastically improves the time to recovery of the DU.

V. EVALUATION

To evaluate the suggested solution, we deploy our prototypical implementation in a 5G standalone testbed as shown

Lisa Wernet, Fridolin Siegmund, Ralf Kundel, Tobias Meuser, Ralf Steinmetz. *Fault-tolerance in Future RAN: Enabling Redundancy for Efficient Recovery of RAN Network Functions.* To appear in the Proceedings of IEEE Conference on Network Function Virtualization and Software Defined Networks 2023, IEEE, 2023.

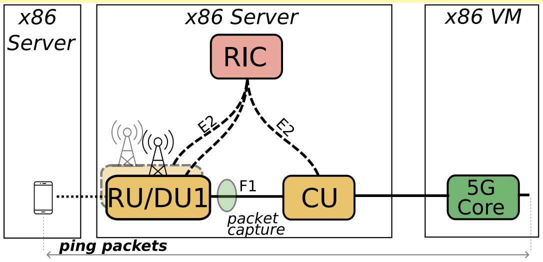


Fig. 5: Evaluation setup for packet capturing.

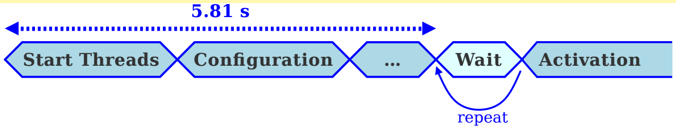


Fig. 6: Time saving in the startup procedure of RU/DU instances in warm standby.

TABLE I: Total lowest measured times for first UE connection attempts T_{ca} for both data sets of 50 connections with and without redundancy.

DU Instances	**Minimum** T_{ca}	**Median** T_{ca}	σ T_{ca}
Single	8.21 s	16.07 s	41.89 s
Redundant	1.9 s	9.46 s	53.07 s
Difference	-6.31 s	-6.61 s	+11.18 s

in Figure 5. The User Equipment (UE) used in our experiments is a Waveshare SIM8202G 5G modem connected to an X86 server. The OpenAirInterface (OAI)-based Radio Access Network (RAN) and the Radio Intelligent Controller (RIC) run on top of an Intel® Core™ i9-9900K CPU with 8 cores, hyperthreading disabled, 32 GB RAM, and one physical 1 Gbit/s Ethernet interface. The radio functionality is built upon the Ettus Research USRP B200 SDR, which is connected via USB to the host system. We use the Free5GC core implementation (version 3.6) running in a Virtual Machine (VM) on a Proxmox Cluster. While the UE and the 5G core remain unmodified, we implemented the RIC and adapted the OAI implementation to allow a fine granular control and monitoring of RU/DU and Central Unit (CU) instances as described in Section IV. Figure 5 also shows the packet capturing point at the F1 interface between CU and RU/DU. For our time measurements, we use the operating system clock, which offers microsecond resolution. This resolution is sufficiently accurate for our purpose since the RAN itself induces delays in the range of milliseconds, affecting our measurements. At the F1 interface, it can be observed if a RU/DU is operating correctly. Connection attempts that the UE initiates are processed by the RU/DU and forwarded to the CU. Only a fully working Distributed Unit (DU) processes and sends such UE messages to the CU. For this evaluation, the `InitialULRRCMessageTransfer` message, indicating an operational DU passing through the Radio Resource Control (RRC) messages from the UE indicating a connection attempt to the CU, is traced at the F1 interface. T_{ca} is the elapsed time between the beginning of a Stream Control Transmission Protocol (SCTP) handshake between RU/DU and CU and the first transmission of an `InitialULRRCMessageTransfer` message at F1 interface. The SCTP handshake indicates that the RU/DU changed from warm standby to operational. If started as cold boot, it indicates the startup of RU/DU.

Figure 6 shows a simplified sequence of tasks done in the parallel threads in the RU/DU. The halting points are placed right before data transmission out of the RU/DU occurs, whether over F1 to the CU or over the radio link. We placed timestamps in the RU/DU code to measure the time between start and halting points. These measurements are used to verify the following measurements at the F1 interface. Until the latest halting point is reached, 5.81 s are passed in average and therefore saved when bringing a RU/DU in warm standby at this halting point to operation compared to a cold start.

A. Single RU/DU instance

To evaluate the decreased disruption time induced by our redundant RU/DU architecture, we first consider the case without any redundancy as benchmark. The RU/DU is started in a scripted loop with the task of automatically restarting the RU/DU in case of a crash or any other disruption where the running code is exiting. Our RIC implementation triggers the RU/DU disruption by terminating the running process. With this procedure, we control the test run time and the number of RU/DU restarts. For the non-redundant case, we collected traces from 50 iterations. In that case, the minimum time to recovery T_{ca} is measured with $8.21s$ as shown in Table I, while the median time to recovery was $16.07s$.

B. Redundant RU/DU instances

In the second case, the RIC controls the halting points in the RU/DU instances and ensures that a redundant RU/DU instance is already in warm standby when taking over the operation of the active RU/DU. Now, the evaluation setup consists of two RU/DU instances. After a disruption, the previously operational RU/DU instance is restarted in warm standby mode. Then, this inactive instance acts as the redundant RU/DU instance. Like in the non-redundant setup, we execute 50 runs and collect 50 data sets.

Table I compares the non-redundant and the redundant measurements. Restarting the same DU without redundancy results in a minimum T_{ca} measured with $8.21s$. The minimum T_{ca} is reduced by $6.31s$ or 76.86% to $1.9s$ when the redundant DU in warm standby takes over operation in case of disruption compared to restarting the same DU again. The median is reduced by 6.61 s or 41.13 % to 9.46 s while the standard deviation is increased by $11.18s$ to $53.07s$. This high standard deviation may be caused by the reaction of the UE to a failed RU/DU, which was visible in some runs that induced high T_{ca} values up to 200 s. The UE behavior in case of a disrupted RU/DU is likely dependent on proprietary implementations at UE side. However, migrating states such as the Radio Network

Temporary Identifier (RNTI) to the redundant DU may be a suitable measure to decrease the standard deviation. This question is left for future research.

The deviation between the median reduction measured at the F1 interface and the software measurement in Figure 6 is only $500\,ms$. This deviation is explainable due to measurement inaccuracy between in code measurements and on wire measurements. Also, the overhead for switching the halting points to operation, i.e., the delay induced by the RIC and the detection time of the disruption, is included in the $500\,ms$.

VI. CONCLUSION AND FUTURE WORK

In a disaggregated and virtualized 5G Radio Access Network (RAN), enhancing the dependability of an individual Virtual Network Function (VNF) increases the overall dependability of the 5G network. Usually, the deployment of VNFs by the Service Management and Orchestration (SMO) framework of the Radio Intelligent Controller (RIC) is an efficient and standardized process. However, in case of an unexpected failure of a network function, the monitoring capabilities of this non-real time framework are not suitable for a timely reaction to failures, and the normal restart of VNFs leads to a relevant disruption.

In this work, we showed that enabling redundancy in combination with warm standby boosts the self-healing capacities of disaggregated RANs. We propose to extend existing service models of the near-real time RIC to support the surveillance of availability and the monitoring of relevant states that describe the faultless function of VNFs. We demonstrate that an efficient recovery mechanism can decrease the total downtime of the RAN by 76.86% (median 41.13%) compared to terminating and restarting faulty instances.

While this work aims at designing and realizing a redundant O-RAN VNF architecture, it also opens up the possibility for extended RIC functionality in the future. For example, RAN VNFs could be operated in hot standby with extended state migration to achieve an additional reduction of downtime. However, for this, efficient mechanisms for this state migration need to be developed. States such as the Radio Network Temporary Identifier (RNTI), General Packet Radio Service Tunneling Protocol for User Plane (GTP-U) Tunnel Endpoint Identifiers (TEIDs) and buffers could be migrated between DU instances. Migrating such states may reduce the high standard deviation we measured in our evaluation. With further reduced downtimes, enhanced evaluation mechanisms will be required. Programmable networking hardware could be used to measure Quality of Service (QoS) metrics inside the network with a highly increased time resolution [16]. State migration is also relevant to assess user service downtime as it allows to reuse the existing Protocol Data Unit (PDU) session. In addition to state migration, extracted states from RAN VNFs could be used to improve failure detection and identification of deviations from the intended operation of the VNFs. For this purpose, the states could be passed to northbound applications in the RIC, e.g., to a xApp for anomaly detection.

ACKNOWLEDGEMENT

This work has been co-funded by the German Research Foundation (DFG) within the Collaborative Research Center (CRC) 1053 MAKI. The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany in the project “Open6GHub” (grant number: 16KISK014).

REFERENCES

[1] A. Khalil, B. Becker, L. Wernet, R. Kundel, B. Richerzhagen, T. Meuser, and R. Steinmetz, “Dependability: Enablers in 5G Campus Networks for Industry 4.0,” in *2023 19th International Conference on the Design of Reliable Communication Networks (DRCN)*. IEEE, Apr 2023.

[2] 3GPP, “Release 16 Description; Summary of Rel-16 Work Items,” 3GPP, techreport TR 21.916 version 16.2.0 Release 16, Jul 2022. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3493

[3] O-RAN Alliance, “O-RAN Working Group 1, Control, User and Synchronization Plane Specification-RAN Architecture Description,” O-RAN Alliance e.V., techreport O-RAN.WG1.OAD-R003-v09.00, Jun 2023. [Online]. Available: https://orandownloadsweb.azurewebsites.net/specifications

[4] M. Polese, L. Bonati, S. D’Oro, S. Basagni, and T. Melodia, “Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges,” *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1411, Jan 2023.

[5] L. M. P. Larsen, A. Checko, and H. L. Christiansen, “A Survey of the Functional Splits Proposed for 5G Mobile Crosshaul Networks,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 146–172, 1. quarter 2019.

[6] A. S. Abdalla, P. S. Upadhyaya, V. K. Shah, and V. Marojevic, “Toward Next Generation Open Radio Access Networks: What O-RAN Can and Cannot Do!” *IEEE Network*, vol. 36, no. 6, pp. 206–213, Nov 2022.

[7] R. Kundel, T. Meuser, T. Koppe, R. Hark, and R. Steinmetz, “User plane hardware acceleration in access networks: Experiences in offloading network functions in real 5g deployments.” in *HICSS*, 2022, pp. 1–10.

[8] W. Y. Yun and J. H. Cha, “Optimal design of a general warm standby system,” *Reliability Engineering & System Safety*, vol. 95, no. 8, pp. 880–886, Aug 2010.

[9] H. Chantre and N. L. S. da Fonseca, “Reliable Broadcasting in 5G NFV-Based Networks,” *IEEE Communications Magazine*, vol. 56, no. 3, pp. 218–224, Mar 2018.

[10] O-RAN Alliance, “O-RAN Working Group 3, Near-Real-time RAN Intelligent Controller, Use Cases and Requirements,” O-RAN Alliance e.V., techreport O-RAN.WG3.UCR-R003-v03.00, Mar 2023. [Online]. Available: https://orandownloadsweb.azurewebsites.net/specifications

[11] R. Schmidt, M. Irazabal, and N. Nikaein, “FlexRIC: an SDK for next-generation SD-RANs,” in *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*. ACM, Dec 2021.

[12] OAI, “OpenAirInterface MOSAIC5G Project Group.” [Online]. Available: https://openairinterface.org/mosaic5g/

[13] S. Ramanathan, A. Bhattacharyya, K. Kondepu, M. Razo, M. Tacca, L. Valcarenghi, and A. Fumagalli, “Demonstration of Containerized Central Unit Live Migration in 5G Radio Access Network,” in *2022 IEEE 8th International Conference on Network Softwarization (NetSoft)*. IEEE, Jun 2022.

[14] A. Leiter, A. Hegyi, N. Galambosi, E. Lami, and P. Fazekas, “Automatic failover of 5G container-based User Plane Function by ONAP closed-loop orchestration,” in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, Apr 2022.

[15] C. Surianarayanan, G. Ganapathy, and P. Raj, *Essentials of Microservices Architecture*. Taylor & Francis, Aug 2019.

[16] R. Kundel, F. Siegmund, R. Hark, A. Rizk, and B. Koldehofe, “Network Testing Utilizing Programmable Network Hardware,” *IEEE Communications Magazine*, vol. 60, no. 2, pp. 12–17, Feb 2022.