Designing Reliable Virtualized Radio Access Networks

Ufuk Usubütün*, André Gomes†‡, Shankaranarayanan Puzhavakath Narayanan§,
Matti Hiltunen§, Shivendra Panwar*

*New York University, NY   †Virginia Tech, VA   ‡Rowan University, NJ   §AT&T Labs Research, NJ

Emails: {usubutun, panwar}@nyu.edu, gomesa@rowan.edu, {snarayanan, hiltunen}@research.att.com

Abstract—As virtualization of Radio Access Networks (RAN) gains momentum, understanding the impact of hardware and software disaggregation on resiliency becomes critical to meet the high availability requirements of mobile networks. Our paper presents an analytical model, using continuous time Markov chains, to study the impact of virtualization and disaggregation on RAN availability. Our evaluation, assuming typical parameter value ranges for failure and recovery rates, points to containerized platform reliability as a constraint on vRAN availability. We also find that with active-passive replication, increasing hardware replication factor beyond 2 may not bring any benefits unless failover times are reduced. We also compare the reliability of centralized and distributed virtualized central units.

Index Terms—Virtual Radio Access Network, availability, replication, reliability, virtualization, disaggregation, modeling

I. INTRODUCTION

Radio Access Network (RAN) resiliency is becoming increasingly critical as we grow more reliant on mobile networks. To provide dependable networks, Mobile Network Operators (MNOs) routinely use advanced network planning strategies like coverage optimization, spectrum redundancy, and optimized scheduling of planned maintenance activities, along with upgrading their network equipment for resiliency. Maintaining or improving availability is critical as MNOs transition to virtualized RAN (vRAN).

The vRAN architecture disaggregates the baseband unit (BBU) into two components: a distributed unit (DU) that is responsible for processing lower network layers (e.g., PHY, MAC) and a centralized unit (CU) that is responsible for processing higher network layers (e.g., PDCP, RRC). Fig. 1 contrasts this architectural change from traditional RAN to vRAN. vRAN also takes the path of virtualization as opposed to traditional BBUs, which rely on purpose-built hardware running highly customized RAN software. In vRAN, the DU and the CU functions are deployed as software applications, often designed as micro-services, that are deployed on containerized cloud platforms (CaaS). These are hosted on commercial off-the-shelf (COTS) servers potentially provided by different vendors. This allows MNOs to leverage increased flexibility
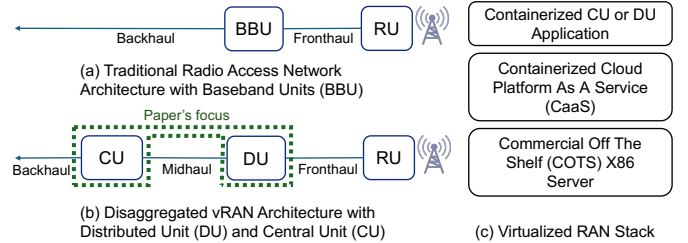
Fig. 1. Traditional vs Virtualized Disaggregated Radio Access Network (RAN) Architectures.

in terms of vendor, network deployment and management options. However, the fundamental architectural changes brought by vRAN require a careful consideration of RAN resiliency.

Prior works have studied RAN resiliency by focusing on the radio link (e.g., through coverage planning [1], spectrum dimensioning [2], and user offloading [3]) and, more recently, on specific vRAN components beyond the radio link (e.g., improving the PHY layer [4] and stackwide [5] DU failovers).

In contrast, our work focuses on examining the architectural change introduced by virtualization and disaggregation, and addresses a critical gap in the study of the reliability of vRANs [6]. We develop analytical models to capture the virtualized and disaggregated vRAN architecture and identify reliability bottlenecks. We use Markov chains to represent the availability-state of each RAN element, including the COTS server, CaaS platform (which we consider jointly with the operating system (OS)) and CU/DU applications. We use time-to-fail and repair statistics, replication factors and failover mechanisms applicable to the respective network elements to model the state transitions. Our models focus only on the functions of the traditional BBU that are disaggregated into CU and DU (highlighted in Fig. 1(b)). We omit the analysis of midhaul as it is often realized through the existing standard transport infrastructure supporting the backhaul. We model both the active-active and active-passive replication and failover modes, and discuss their impact on availability. We also analyze the impact on overall availability when the CU is centralized, serving multiple cell-sites.

To the best of our knowledge, this is the first paper to develop availability models capturing the key elements of vRAN. Our evaluation of the analytical model shows that: (i) OS/CaaS platform reliability is likely to be a key bottle-
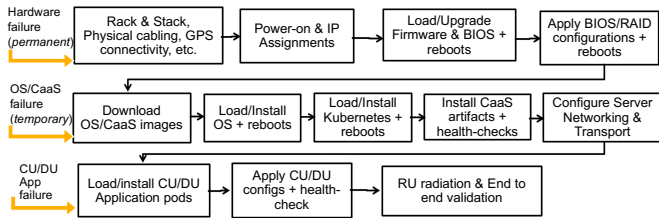
Fig. 2. Recovering from failures in a vRAN.

(a) A cluster without platform replication   (b) A cluster with *active-active platform* replication   (c) A cluster with *active-passive platform* replication

Fig. 3. Different modes of platform replication. Note that in our models we assume that OS/CaaS software is not independently replicated.

neck, and improving it is critical to achieving higher vRAN availability, (ii) in active-passive replicated systems, increasing hardware replication factor beyond 2 provides only limited benefits as failover times tend to limit availability and, (iii) centralized CU deployments have the potential to provide high availability as long as the CU availability is carefully calibrated to reduce single points of failure.

II. BACKGROUND

This section outlines the types of failures that may affect vRANs and potential ways of mitigating their impact.

A. Nature of Failures and Recovery in vRAN

The virtualized nature of vRAN requires different recovery actions depending on the nature of the failure. Figure 2 shows the steps involved in replacing a failed server at the cell-sites – from physically connecting the server to end-to-end validation that checks if the RAN is ready to serve traffic. Note that some recovery actions may skip certain steps, and each step can take from a few seconds (e.g., applying configurations) up to tens of minutes (e.g., loading BIOS/firmware and multiple reboots).

For simplicity, we group the failures into three categories: (i) hardware (that we term *permanent*), (ii) OS/CaaS software (that we term *temporary*), and (iii) CU/DU application failures. The duration of hardware failure outages are dominated by a team having to drive to the location of failure and are usually resolved within roughly 10 hours. For non-permanent failures, recovery time is dominated by the steps outlined in Fig. 2. We expect its duration to be of the order of tens of minutes for OS/CaaS (*temporary*) failures and of the order of minutes for CU/DU application failures. However, as data on commercial implementations of vRAN is not available to date to provide precise numbers, we consider wide ranges of values when we evaluate the dynamics of recovering from OS/CaaS software or CU/DU application failures, with specific ranges presented in § IV.

B. Replication of RAN Elements for Resiliency

vRAN decouples CU/DU applications and the hardware platform, enabling them to be replicated for both resiliency and scaling their capacity (e.g., Fig. 3). This allows both the CU and DU to have one or more application instances running on one or more servers. While many combinations of replication are possible, we focus on two commonly adopted replication models: (i) one or more servers with a single CU/DU application instance on each server and (ii) more than one application instance on each server, reflecting the scenario
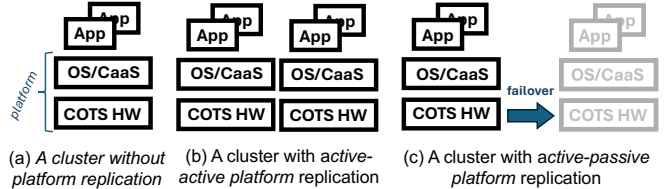
where the CU/DU application instances serve different carrier configurations (e.g., mid-band vs. low-band).

Finally, the replicas of the *platform* (defined in the next section along with *cluster*) can be either in:
Active-Active mode: Here the incoming traffic to a network function can be processed by any of the replicas independently; therefore the function would be assumed unavailable only when all of its replicas are unavailable.
Active-Passive mode: Here the incoming traffic coming to a network function would be processed only by a single active replica, with the other passive replicas remaining in a stand-by or idle state. A fail-over mechanism detects the failure of the active replica and elects one of the passive replicas to take over as the active replica [5]. Therefore, the failure of an active replica will render the network function unavailable until the failover mechanism transitions the function to a previously passive replica. The failure of a passive replica will not lead to an outage.

III. MODELING vRAN AVAILABILITY

In this section we first provide necessary definitions and assumptions for modeling. We then model the availability of a DU/CU. We finally consider the reliability implications of a centralized vs. distributed CU.

A. Definitions

We quantify network resiliency through **availability**, i.e., the fraction of time a network is ready to serve incoming network traffic. We characterize failure dynamics using the standard metrics: **mean time-to-failure (MTTF)**, **mean time-to-repair (MTTR)** [7], and a metric called **mean time-to-failover (MTFO)** that we define as the average time it takes for a system to failover from a failed replica to a functioning one. In this work, we assume that the time-to-failure, time-to-recover and time-to-failover are exponentially distributed. This assumption allows us to build our models using continuous time Markov chains (CTMC) inspired by classic machine-repairman problems [8].

We define a **platform** to refer to the combination of a COTS server and an OS/CaaS environment. We then define a **cluster** to contain all replicas of a platform with all CU/DU application replicas running on top of them. Fig. 3 demonstrates three examples of clusters without and with different modes of platform replication. Therefore, the CU and DU shown in Fig. 1(b) represent by one distinct cluster each in our terminology.

We initially set up models to calculate the availability of platforms and CU/DU applications separately. We then
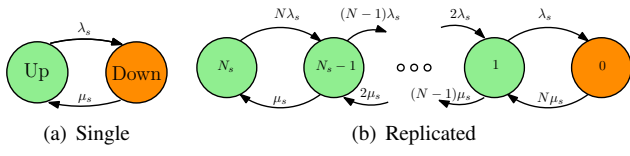
Fig. 4. CU/DU application availability

combine the two models to obtain the availability of a cluster. We finally evaluate the availability of the vRAN by taking into account the geographical placement of the CU and DU and obtain a metric representing the number of unavailable cell-sites.

B. Modeling CU/DU Application Availability

Let the MTTF and MTTR for CU/DU application be $\text{MTTF}_s = 1/\lambda_s$ and $\text{MTTR}_s = 1/\mu_s$. Note that, for convenience, all notation used in this paper is listed in Table I. We model a single, non-replicated, CU/DU application using the two state CTMC in Fig. 4(a). The stationary probabilities of the two states are as follows [8]:

$$\text{Pr}(\text{Up}) = \frac{\mu_s}{\lambda_s + \mu_s}, \quad \text{Pr}(\text{Down}) = \frac{\lambda_s}{\lambda_s + \mu_s}. \quad (1)$$

We assume that vRAN applications are built to be cloud-native and are deployed in virtualized environments where a controller is in charge of detecting failures and restarting software instances. This environment allows us to assume that all replicas of applications fail and recover independently. We therefore formulate replicated applications as N_s independent systems, where N_s is the replication factor, as shown in Figure 4(b), where the states represent the number of up replicas. This yields a binomial distribution p_i of having $i \in \{0, 1, \ldots, N_s\}$ replicas up, $p_i = \binom{N_s}{i} \left(\frac{\mu_s}{\lambda_s + \mu_s}\right)^i \left(\frac{\lambda_s}{\lambda_s + \mu_s}\right)^{N_s - i}$ and the system stays available as long as there is at least one available replica [9]. Therefore, the availability of a CU/DU application with replication factor N_s is equal to $f_s(N_s) = 1 - p_0$.

C. Modeling Platform Availability

As introduced in §II-A, the platform can suffer both from temporary and permanent failures. As the recovery dynamics of the two types of outages are related, we model them together. Let the MTTF and MTTR of temporary failures be $\text{MTTF}_o = 1/\lambda_o$ and $\text{MTTR}_o = 1/\mu_o$ and that of permanent failures be $\text{MTTF}_h = 1/\lambda_h$ and $\text{MTTR}_h = 1/\mu_h$, respectively. Fig. 5(a) illustrates the dynamics of the non-replicated platform: temporary and permanent failures occur and are repaired at their own respective rates. Note that the model also shows the case when a permanent failure occurs during a temporary failure with the downward transition.

To represent cases with replication, we provide different models for active-active and active-passive modes:
Active-active replication of the platform: As any replica can be involved in both temporary and permanent failures with different recovery dynamics, states capturing all combinations of occurences of such outages are needed. The set $\mathcal{S}_A = \{a^b : \forall a, b \in \mathbb{Z}^+, a + b \leq N_h\}$ contains all states for a replication
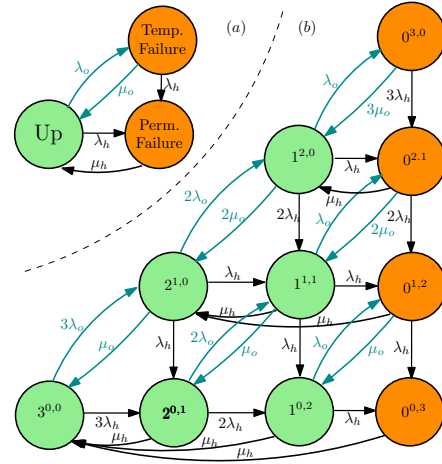

Fig. 5. Platform with (a) no replication and (b) *active-active* replication, $N_h = 3$.

factor of N_h and where $\mathbb{Z}^+$ stands for non-negative integers. In the figures, a third redundant index, c, is introduced to make it easier to interpret the states $a^{b,c}$, as explained next. Index a represents the number of functional replicas, b is the number of replicas with temporary failures and c is the the number of those with permanent failures. The third index c can be found by solving for $a + b + c = N_h$. In Figure 5(b), the rates $r(., .)$ between pairs of state are as follows:

$$r(a^b, (a-1)^{b+1}) = a\lambda_o \quad (2)$$
$$r(a^b, (a+1)^{b-1}) = b\mu_o \quad (3)$$
$$r(a^b, (a-1)^b) = a\lambda_h \quad (4)$$
$$r(a^b, a^{b-1}) = b\lambda_h \quad (5)$$
$$r(a^b, (N_h - b)^b) = \mu_h \quad , \text{ if } a + b < N_h. \quad (6)$$

Observe that the rates of failures are scaled by the number of replicas that are functional (index a), which can experience failure (Eq. 2,4), and that the rates of recovery from temporary failures are likewise scaled by the number of replicas that are experiencing a temporary outage (index b) (Eq. 3). Likewise, downward permanent failure transitions are also scaled by the number of replicas in temporary outage (Eq. 5). Finally, we assumed that the repair times for permanent, i.e., hardware, failures are dominated by the time it takes for a service team to be dispatched to the location of the failure. Therefore, all such repair transitions lead to the respective state where no permanent failures remain (Eq. 6).

We denote the steady state probabilities $\pi(s)$ for each state $s \in \mathcal{S}_A$. For a given state space and its infinitesimal generator matrix $\mathbf{Q}$, one can find the embedded discrete time Markov chain (DTMC), solve the stationary state probabilities of the DTMC numerically and scale the solution back such that it provides the steady state solution to the original CTMC [10]. Using this standard procedure, we populate $\mathbf{Q}$ and solve for the stationary probabilities $\pi(s)$.

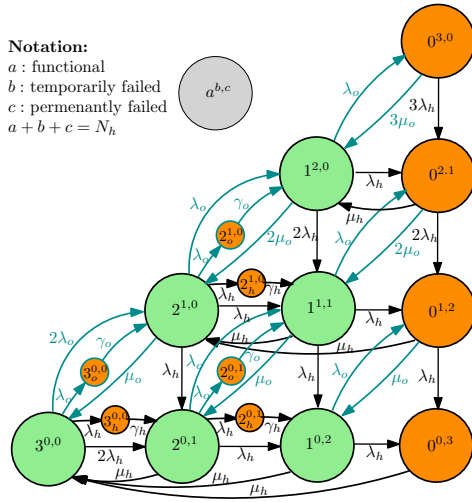The probability of outage p_a can then be found by summing over the steady state probabilities of outage states:

Fig. 6. *Active-passive replicated platform, $N_h = 3$.*

$\sum_{b=0}^{N_h} \pi(0^b) = p_a$. Therefore, the availability of an active-active replicated platform with replication factor N_h is equal to $f_{p,a}(N_h) = 1 - p_a$.

Active-passive replication of the platform: To model failovers in the active-passive case, let the MTFO for temporary and permanent failures be $\text{MTFO}_o = 1/\gamma_o$ and $\text{MTFO}_h = 1/\gamma_h$ respectively. We introduce new failover states a_o^b and a_h^b in order to represent the new states during failovers. Formally, the set $\mathcal{S}_F = \{a_k^b : \forall k \in \{o, h\}, \forall a, b \in \mathbb{Z}^+, 2 \leq a, a + b \leq N_h\}$ contains all new states. Therefore $\mathcal{S}_P = \mathcal{S}_A \cup \mathcal{S}_F$ contains all states for the active-passive model. Fig. 6 depicts the model with the same redundant index c. The model accommodates one major change: Only one of the functioning replicas is actively serving at a time, while others are in standby mode. Therefore, a failure at an active replica triggers a failover that leads to temporary outage. However, a failure in a passive, yet functional, replica does not cause an outage but only reduces the number of available functional components. Finally, there can be no failover if there are no functional passive replicas left. State transition rate equations describing the model are as follows:

$$r(a^b, (a-1)^{b+1}) = \max\{a-1, 1\}\lambda_o \quad (7)$$

$$r(a^b, a_o^b) = \lambda_o, \quad r(a_o^b, (a-1)^b) = \gamma_o \quad (8)$$

$$r(a^b, (a-1)^b) = \max\{a-1, 1\}\lambda_h \quad (9)$$

$$r(a^b, a_h^b) = \lambda_h, \quad r(a_h^b, (a-1)^b) = \gamma_h \quad (10)$$

Equations 3, 4 and 6 remain unchanged from the active-active case and also describe this model. In line with the changes discussed, equations 7 and 9 capture the failures incurred by passive replicas, whereas equations 8 and 10 capture failures incurred by the serving replica and its failover procedure.

The stationary probabilities $\pi(s)$ can be found using the the same methodology as described for the active-active case for all $s \in \mathcal{S}_P$. This time, the outage will be the sum of states with no surviving replicas and the failover states, i.e., $p_p = \sum_{b=0}^{N_h} \pi(0^b) + \sum_{s \in \mathcal{S}_F} \pi(s)$ and the availability will be $f_{p,p}(N_h) = 1 - p_p$.
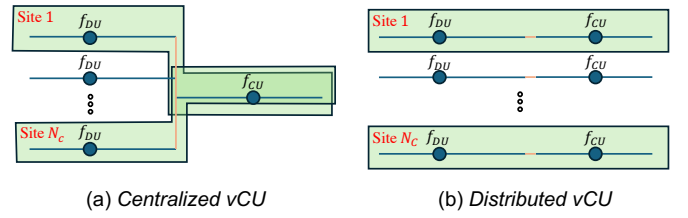

(a) *Centralized vCU*   (b) *Distributed vCU*

Fig. 7. Cell Sites Dependencies with respect to CU placement.

D. Modeling Cluster Availability

We obtain cluster availabilities $f_{c,a}$ and $f_{c,p}$ for a cluster with active-active or active-passive platform replication as:

$$f_{c,a} = f_{p,a}(N_h) \cdot f_s(N_s \cdot N_h) \quad (11)$$

$$f_{c,p} = f_{p,p}(N_h) \cdot f_s(N_s) \quad (12)$$

In case of active-active platform replication, we assume application instances across different platforms can coordinate, therefore the number of application replicas are scaled by the number of platform replicas. This is visually illustrated in Fig. 3. The availability of the CU (f_{CU}) and DU (f_{DU}) can now be calculated using one of the two options for cluster availability above in equations 11 and 12.

E. Modeling vRAN Availability

The disaggregated vRAN architecture allows the CU and DU to be either co-located or be placed at different physical locations. Here, we focus on two cases of centralized and distributed CUs. Having a centralized CU indicates that multiple DUs at different locations would be served by that single centralized CU, as shown in Fig. 7. Let us compare two deployments with N_c cell sites with and without a centralized CU. We assume independence between all instances of DUs and CUs. Let X_c and X_d be the discrete random variables representing the number of unavailable cell sites for centralized and distributed scenarios respectively.

$$\Pr(X_c = k) = \begin{cases} f_{CU}\binom{N_c}{k}(\overline{f_{DU}})^k(f_{DU})^{N_c-k} \text{, else} \\ \overline{f_{CU}} + f_{CU}(\overline{f_{DU}})^{N_c} \quad \text{, if } k = N_c \end{cases} \quad (13)$$

$$\Pr(X_d = k) = \binom{N_c}{k}(1 - f_{DU}f_{CU})^k (f_{DU}f_{CU})^{N_c-k} \quad (14)$$

Interestingly, even though the two distributions above are very different, the expected number of unavailable cells is the same for both, as derived in Appendix A: $\mathbb{E}[X_d] = \mathbb{E}[X_c] = N_c(1 - f_{DU}f_{CU})$. This can be intuitively explained as follows. The probability of no outage for the centralized case $\Pr(X_c = 0)$ is higher than for the distributed case $\Pr(X_d = 0)$. However, when the centralized CU does fail, it brings down all the cells it is connected to. This is unlike the failure of a distributed CU deployment for which the failure of all N_c cells is extremely unlikely. Note that the outage probability of a cell is also identical in both cases and is equal to $1 - f_{DU}f_{CU}$.

Term	Definition	Dependence
$1/\lambda_s, 1/\mu_s$	CU/DU Application Mean-Time-to-Fail / Mean-Time-to-Recover	-
$1/\lambda_o, 1/\mu_o$	OS/CaaS Software Mean-Time-to-Fail / Mean-Time-to-Recover	-
$1/\lambda_h, 1/\mu_h$	Hardware Mean-Time-to-Fail / Mean-Time-to-Recover	-
$1/\gamma_h$	Hardware Mean-Time-to-Failover	-
$1/\gamma_o$	OS/CaaS Software Mean-Time-to-Failover	-
N_s, N_h	Replication Factor for CU/DU Application / Hardware	-
f_s	CU/DU Application Availability	λ_s, μ_s, N_s
$f_{p,a}$	Platform Availability with *Active-Active* Replication	$\lambda_o, \mu_o, \lambda_h, \mu_h, N_h$
$f_{p,p}$	Platform Availability with *Active-Passive* Replication	$\lambda_o, \mu_o, \lambda_h, \mu_h, N_h, \gamma_o, \gamma_h$
$f_{c,a}$	Cluster Availability with *Active-Active* Replicated Platform	$f_{p,a}, f_s$
$f_{c,p}$	Cluster Availability with *Active-Passive* Replicated Platform	$f_{p,p}, f_s$
f_{CU}, f_{DU}	Availability of CU / DU	$f_{c,a}$ or $f_{c,p}$
N_c	Number of Cell Sites	-
X_c, X_d	Number of Unavailable Cell Sites in Centralized / Distributed CU Settings	f_{DU}, f_{CU}, N_c

Parameter	Value	Reference
MTTF_h	12 - 33 years (3% - 8%)*	[11]–[13]
MTTR_h	10 hours	§II-A
MTTF_o	17 - 70 days	[14, Table 10]
MTTR_o	0.5 min - 1.5 hours	[15], §II-A
MTFO	0.5 - 10 min	§II-A
MTTF_s	7 - 52 days	[16]
MTTR_s	1s - 30 min	[17]–[19], §II-A

* shows the equivalent annualized failure rate for hardware. The MTFO_o
and MTFO_h values used are the same and correspond to the value shown
for MTFO in this table.

IV. EVALUATION AND RESULTS

This section reports on vRAN availability using the models
introduced in § III. For the evaluation, we consider the values
in Table II to assess platform, cluster, and vRAN availability.
Given the current absence of studies reporting on the failure
and recovery rates of vRAN implementations, we refer to
typical value ranges from related domains (see Table II for
references) and evaluate the impact of these parameter values
on the vRAN availability. We evaluate our models with a range
of parameter values to reflect the variable duration of failure
and recovery times discussed in § II-A.

A. Platform Availability

We start with the platform availability (as modeled in
§III-C) in an active-active replication scenario. In Fig. 8(a),
we explore the effect of different combinations of MTTR_o and
MTTF_o for fixed hardware failure and repair rates to investi-
gate the impact of OS/CaaS outages on the platform availabil-
ity. Observe that platform availability does not improve further
after $\text{MTTF}_o > 10^3$ days. However, in practice, OS/CaaS can
have lower MTTF_o (e.g., [14]), which implies that platform
availability can be significantly impacted by failures in the
OS/CaaS. In such cases, timely recovery ($\text{MTTR}_o \in \{1, 15\}$
min) can significantly mitigate the impact of failures, reducing
the observed platform outage probability.

Fig. 8(b) considers the unfavorable scenario of
$\text{MTTR}_o = 1.5$ hours to represent longer repair times
that may involve reinstalling OS/CaaS environments.
Interestingly, for $\text{MTTF}_o \leq 10^2$ days, Fig. 8(b) shows
little benefit from using more reliable hardware (i.e., higher
MTTF_h) as OS/CaaS failures dominate the overall outage
probability. Hardware replication, which also replicates
OS/CaaS as each server has its own OS/CaaS environment,
can compensate for less reliable OS/CaaS environments at the
expense of additional hardware and potentially more OS/CaaS
software licenses. These results highlight the relevance of
carefully selecting reliable OS/CaaS environments to avoid
frequent failures that can limit vRAN availability.

In Fig. 8(c), we consider a counterpart scenario with active-
passive replication to study the impact of failover mechanisms
on the platform availability. Notice that there is no failover if
$N_h = 1$ (i.e., no replication). Interestingly, we observe that
the outage probability is insensitive to additional replication
beyond $N_h = 2$ for the region MTFO > 10 sec. This suggests
that investments in fast failover (such as the mechanisms
proposed in [4], [5]) can significantly improve availability. It
also shows that when failover is slow, investment in additional
hardware replication does not lead to increased availability.

B. Cluster Availability

Table IV reports on the entire cluster availability (i.e.,
platform plus CU/DU applications, calculated as in § III-D)
for an active-active replication setting. For ease of visual-
ization, we focus on the parameter combinations that lead
to six-9s availability, where six-9s means availability in the
range $[0.999999, 0.9999999)$. We present the overall availabil-
ity (#9s) as well as the corresponding platform (#9s$_p$) and
OS/CaaS (#9s$_s$) availabilities. For the range of parameter val-
ues in consideration, the target availability of six-9s can only
be achieved by hardware replication (i.e., $N_h \geq 2$). After that,
software becomes the availability bottleneck, requiring more
reliable OS/CaaS environments (e.g., $\text{MTTF}_o = 10$ months
needed to achieve seven-9s) and faster CU/DU recovery mech-
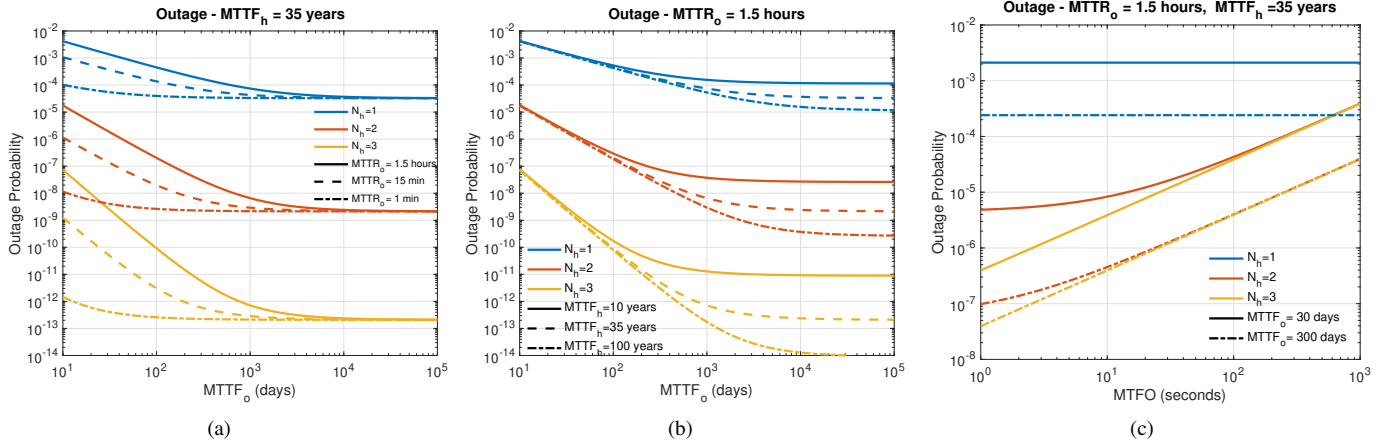anisms (e.g., $\text{MTTR}_s = 5$ min needed to meet seven-9s).

Fig. 8. Platform availability in (a) active-active replication with $\text{MTTF}_h = 35$ years, (b) active-active replication with $\text{MTTR}_o = 1.5$ h, and (c) active-passive replication with $\text{MTTR}_o = 1.5$ h and $\text{MTTF}_h = 35$ years. $\text{MTTR}_h = 10$ hours for all scenarios.

TABLE III
PROBABILITY OF CELL OUTAGE AND PROBABILITY OF ALL CELLS
UNAVAILABLE AS CU OUTAGE PROBABILITY IS DECREASED

DU Outage $1 - f_{DU}$	CU Outage $1 - f_{CU}$	Cell Outage $1 - f_{DU} f_{CU}$	All Cells Unavailable $\Pr(X_c = N_c)$
10^{-5}	10^{-5}	$\sim 1.99 \cdot 10^{-5}$	$\sim 10^{-5}$
10^{-5}	10^{-6}	$\sim 1.10 \cdot 10^{-5}$	$\sim 10^{-6}$
10^{-5}	10^{-7}	$\sim 1.01 \cdot 10^{-5}$	$\sim 10^{-7}$

Similarly, Table V shows the cluster availability for an active-passive scenario. We consider MTFO values of $\{10, 100\}$ sec. Notice, however, that six-9s availability is only achievable with an MTFO of 10 sec even in settings with more reliable hardware, better OS/CaaS recovery times, or replication (see Table V, rows 3-4). This highlights, again, the importance of investing in fast failover mechanisms to achieve high availability in deployments with active-passive replication. We also see that more CU/DU application replicas (N_s) are needed on each platform in this case, as replicas hosted on passive platform replicas are not immediately available. Note that our model does not capture failovers that could be triggered after all CU/DU application replicas on a platform fail. Therefore, for the case when the MTFO is shorter than the MTTR_s, our model might provide an underestimate of application availability.

C. Assessing Network-wide vRAN Availability

We have found, in §III-E, that the expected number of cells in outage for both centralized and distributed CU cases to be the same and equal to $N_c(1 - f_{DU} f_{CU})$. At the same time, we also found a significant probability for all cells to simultaneously experience an outage for the centralized CU case. To mitigate this, we reason that since it is likely for centralized CUs to be well provisioned to handle the large volume of traffic resulting from a centralized architecture, it can also be cost-effectively designed for higher reliability as compared to a CU in a distributed architecture that serves only one cell. In Table III we consider three scenarios where we increase the CU availability in each successive row. We

provide the cell outage probability, which is the same for both centralized and distributed CU cases, and the probability of all cells being unavailable for the centralized CU case (for $N_c > 2$).

We see that improvements in the CU availability leads to a corresponding reduction in the probability of all cells being unavailable. If the likelihood of this type of outage is an important concern, then the design of a sufficiently reliable centralized CU can mitigate this issue.

V. CONCLUSIONS

Based on our analysis, we came to three main conclusions about vRAN reliability:

- Hardware replication is necessary to achieve higher than five-9s availability.
- OS/CaaS reliability constrains the availability of the vRAN. Therefore increasing its availability and reducing repair times will improve overall vRAN availability
- For active-passive systems, we determined that failover times constrain their reliability.

Our analysis also exposed an interesting characteristic of centralized CU deployments: they are more likely to have all cell sites unavailable than distributed CUs, even while the two have the same average cell availability. This weakness can be mitigated by making centralized CUs more reliable, an investment that can be justified given that there are far fewer CUs in a centralized architecture as compared to a distributed architecture.

REFERENCES

[1] J. G. Andrews, F. Baccelli, and R. K. Ganti, "A tractable approach to coverage and rate in cellular networks," *IEEE Trans. Commun.*, 2011.

[2] A. Gomes, J. Kibiłda, and L. A. DaSilva, "Assessing the spectrum needs for network-wide ultra-reliable communication with meta distributions," *IEEE Commun. Lett.*, 2023.

[3] S. Yang, Y. He, Z. Ge, D. Wang, and J. Xu, "Predictive impact analysis for designing a resilient cellular backhaul network," *Proc. ACM Meas. Anal. Comput. Syst.*, 2017.

[4] N. Lazarev, T. Ji, A. Kalia, D. Kim, I. Marinos, F. Y. Yan, C. Delimitrou, Z. Zhang, and A. Akella, "Resilient baseband processing in virtualized RANs with Slingshot," in *ACM SIGCOMM 2023*, 2023.

TABLE IV
CLUSTER AVAILABILITY IN ACTIVE-ACTIVE REDUNDANCY FOR DIFFERENT COMBINATIONS OF PARAMETERS

#9s	#9s$_p$	#9s$_s$	N_h	MTTF$_h$	MTTF$_o$	MTTR$_o$	N_s	MTTR$_s$
6	6	6	2	10 years	10 months	90 min	1	30 min
6	6	6	2	10 years, 100 years	1 months	15 min	1	30 min
6	6	8	2	10 years	10 months	90 min	1	5 min
6	6	8	2	10 years, 100 years	1 months	15 min	1	5 min
7	7	8	2	10 years	10 months	15 min	1	5 min
7	7	8	2	100 years	10 months	90 min	1	5 min

For all cases MTTR$_h$ = 10 hours, MTTF$_s$ = 2 months.
All possible combinations of parameters in a given row result in the same number of nines for the cluster, platform and application respectively.

TABLE V
CLUSTER AVAILABILITY IN ACTIVE-PASSIVE REDUNDANCY FOR DIFFERENT COMBINATIONS OF PARAMETERS

#9s	#9s$_p$	#9s$_s$	N_h	MTTF$_h$	MTTF$_o$	MTTR$_o$	MTFO	N_s	MTTR$_s$
5	5	5	2, 3	10 years, 100 years	10 months	1 min, 15 min, 90 min	100 sec	2	30 min
5	5	7	2, 3	10 years, 100 years	10 months	1 min, 15 min, 90 min	100 sec	2	5 min
6	6	7	2, 3	10 years, 100 years	10 months	1 min, 15 min, 90 min	10 sec	2	5 min
6	6	8	2, 3	10 years, 100 years	10 months	1 min, 15 min, 90 min	10 sec	3	30 min

For all cases MTTR$_h$ = 10 hours, MTTF$_s$ = 2 months.
All possible combinations of parameters in a given row result in the same number of nines for the cluster, platform and application respectively.

[5] J. Xing, J. Gong, X. Foukas, A. Kalia, D. Kim, and M. Kotaru, *Enabling Resilience in Virtualized RANs with Atlas*. ACM Mobicom, 2023.

[6] D. Tipper, A. Babay, B. Palanisamy, and P. Krishnamurthy, "Network connectivity resilience in next generation backhaul networks: Challenges and future opportunities," *IEEE Trans. Netw. Serv. Manag.*, 2024.

[7] A. Birolini, *Reliability Engineering: Theory and Practice*, ser. Engineering online library. Springer Berlin Heidelberg, 2003.

[8] K. S. Trivedi, *Probability and statistics with reliability, queuing and computer science applications*, 2nd ed. GBR: Wiley, 2001.

[9] C. Weinstein, "Fractional speech loss and talker activity model for TASI and for packet-switched speech," *IEEE Trans. Commun.*, vol. 26, no. 8, pp. 1253–1257, 1978.

[10] R. Howard, *Dynamic Probabilistic Systems Vol II: Semi-Markov and Decision Processes*. Dover, 2007.

[11] J. Scaramella, M. Marden, J. Daly, and R. Perry, "The cost of retaining aging it infrastructure," Technical report, International Data Corporation (IDC), Framingham, MA, Tech. Rep., 2014.

[12] K. V. Vishwanath and N. Nagappan, "Characterizing cloud computing hardware reliability," in *ACM SoCC*, 2010.

[13] G. Wang, L. Zhang, and W. Xu, "What can we learn from four years of data center hardware failures?" in *IEEE/IFIP DSN 2017*, 2017.

[14] R. Matias, M. Prince, L. Borges, C. Sousa, and L. Henrique, "An empirical exploratory study on operating system reliability," in *Proc. ACM Symp. Appl. Comput.*, 2014, pp. 1523–1528.

[15] S. I. Abrita, M. Sarker, F. Abrar, and M. A. Adnan, "Benchmarking VM startup time in the cloud," in *Benchmarking, Measuring, and Optimizing*, C. Zheng and J. Zhan, Eds. Cham: Springer, 2019, pp. 53–64.

[16] P. Jalote, B. Murphy, M. Garzia, B. Errez, and O. R. Way, "Measuring reliability of software products," in *IEEE ISSRE*, 2004.

[17] E. F. Boza, C. L. Abad, S. P. Narayanan, B. Balasubramanian, and M. Jang, "A case for performance-aware deployment of containers," in *WOC*, 2019, pp. 25–30.

[18] T. K. Authors, "Configure liveness, readiness and startup probes," www.kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-startup-probes, accessed: 2024-03-17.

[19] R.-S. Schmoll, T. Fischer, H. Salah, and F. H. P. Fitzek, "Comparing and evaluating application-specific boot times of virtualized instances," in *IEEE 5GWF*, 2019, pp. 602–606.

APPENDIX A
DERIVATION OF THE EXPECTED NUMBER OF UNAVAILABLE CELLS FOR DISTRIBUTED AND CENTRALIZED CUS

A. The Expected Value of X_d:

By definition of the binomial distribution:

$$\mathbb{E}[X_d] = N_c(1 - f_{DU}f_{CU}). \tag{15}$$

B. The Expected Value of X_c:

For simplicity define:

$$f_{CU} = \alpha, \qquad \overline{f_{CU}} = 1 - f_{CU} = \beta. \tag{16}$$

$$f_{DU} = p, \qquad \overline{f_{DU}} = 1 - f_{DU} = q. \tag{17}$$

Substitute α, β, p and q into $\Pr(X_c)$:

$$\Pr(X_c = k) = \begin{cases} \beta + \alpha q^{N_c}, & \text{if } k = N_c, \\ \alpha \binom{N_c}{k} q^k p^{N_c - k}, & \text{else.} \end{cases} \tag{18}$$

Define random variable X' such that,

$$\Pr(X' = k) = \binom{N_c}{k}(q)^k(p)^{N_c - k}. \tag{19}$$

By definition of the binomial random variable

$$\mathbb{E}[X'] = N_c q. \tag{20}$$

We can express $\mathbb{E}[X_c]$ in terms of X' as follows:

$$\mathbb{E}[X_c] = \alpha\big[\mathbb{E}[X'] - N_c \cdot \Pr(X' = N_c)\big] + N_c \cdot \Pr(X_c = N_c) \tag{21}$$

$$= \alpha\big[N_c q - N_c q^{N_c}\big] + N_c \cdot (\beta + \alpha q^{N_c}) \tag{22}$$

$$= N_c(\alpha q + \beta) \tag{23}$$

$$= N_c(\alpha(1 - p) + 1 - \alpha) \tag{24}$$

$$= N_c(1 - p\alpha) \tag{25}$$

$$= N_c(1 - f_{DU}f_{CU}) = \mathbb{E}[X_d]. \tag{26}$$