FUSION: Predictive Reliability–Performance Analytics for Open RAN Systems

Kazu Okumoto
Sakura Software Solutions, LLC

Abstract

This paper proposes a reliability–performance-integrated model for Open RAN systems that explicitly incorporates Kubernetes as a shared-infrastructure constraint. Conventional reliability analysis based on k-of-n structures primarily focuses on component failures and does not adequately capture performance degradation caused by capacity limitations under dynamic traffic conditions.

In the proposed model, system stability is formulated as a function of both structural reliability and effective processing capacity. Kubernetes is modeled not merely as a reliability element but as a capacity ceiling that constrains higher-layer functions such as vDU and vCU. The effective capacity of each layer is defined as the minimum of its intrinsic processing capability and the available infrastructure capacity. System utilization is then expressed as the ratio of traffic load to the minimum effective capacity across infrastructure and application layers.

Analytical results show that system instability can occur due to capacity saturation at bottleneck layers even when structural redundancy conditions are satisfied. The proposed framework provides a quantitative basis for evaluating system-level stability and identifying bottlenecks in Open RAN environments under dynamic operating conditions.

Index Terms

Open RAN, Kubernetes, Reliability–Performance Modeling, Cloud-Native Networks, System Stability

I. Introduction

Open Radio Access Network (Open RAN) has emerged as a key architectural paradigm for next-generation mobile communication systems, enabling functional disaggregation, virtualization, and cloud-native deployment [1], [2]. Traditional base station functions are decomposed into modular components such as the Radio Unit (RU), Distributed Unit (DU), and Centralized Unit (CU), which can be flexibly deployed across edge and regional cloud environments [3].

As shown in Fig. 1, these functional components are deployed on cloud-native platforms (O-Cloud) and orchestrated using systems such as Kubernetes [4], [5]. The Service Management

and Orchestration (SMO) layer provides integrated control and monitoring across the system. While this architecture improves scalability and flexibility, it introduces new challenges in understanding system-level reliability and performance under dynamic operating conditions.

Conventional reliability analysis focuses primarily on component failures and redundancy structures (k-of-n) [6]. However, in modern Open RAN systems, system stability depends not only on structural redundancy but also on the underlying infrastructure's ability to sustain processing loads.

This paper addresses this gap by proposing a unified framework that integrates structural reliability and capacity constraints into a single analytical model. A key contribution is modeling Kubernetes as a shared infrastructure resource that limits system capacity rather than as a simple reliability component.

From a practical perspective, operators and system designers frequently face questions such as:

• Which component becomes the bottleneck under increasing traffic?

• Why does system performance degrade even when redundancy conditions are satisfied?

• How can infrastructure constraints, such as Kubernetes resource limits, be quantitatively evaluated?

• Why do discrepancies arise between test results and real operational behavior?

These questions highlight the need for a framework that can reveal hidden performance constraints and their impact on system stability, which conventional reliability models do not capture.

II. Background

A. Open RAN Architecture

As illustrated in Fig. 1, Open RAN systems consist of:

- **O-RU (Radio Unit):** RF processing and antenna interface
- **O-DU (Distributed Unit):** Real-time baseband processing
- **O-CU (Centralized Unit):** Higher-layer protocol processing

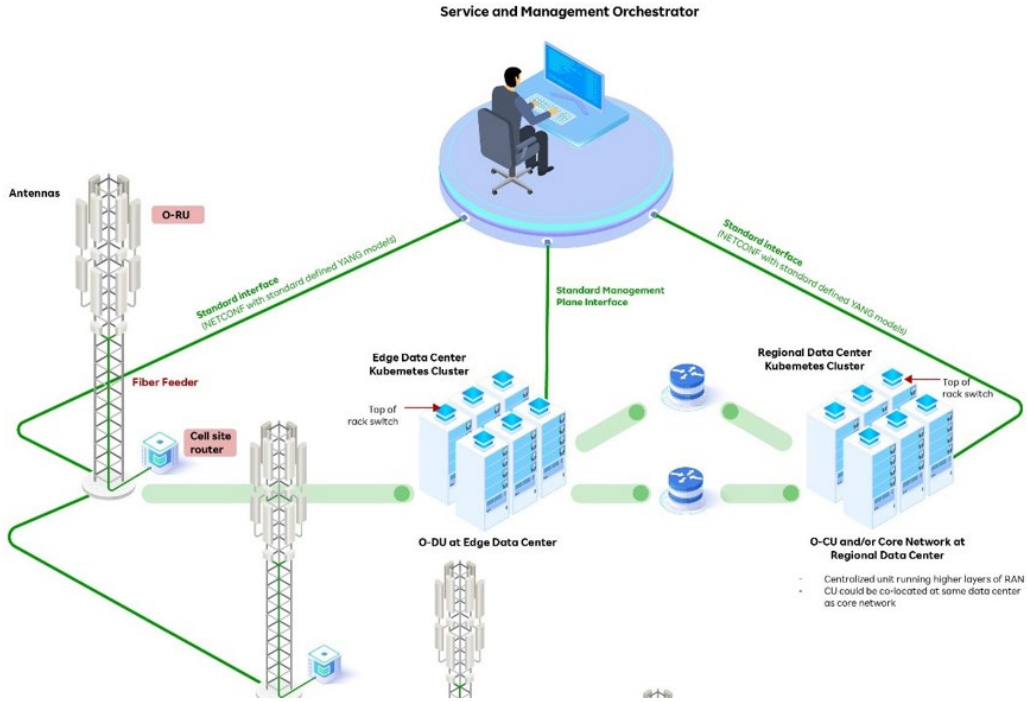

Figure 1. Open RAN Architecture – System View

These components are interconnected via standardized interfaces and deployed across distributed cloud environments.

B. Functional Architecture

Fig. 2 shows a cloud-native implementation where:

- vDU and vCU run as virtualized functions on Kubernetes clusters
- Infrastructure resources (compute, storage, networking) are managed via O-Cloud
- SMO provides orchestration and lifecycle management

This architecture introduces strong dependencies between application performance and infrastructure capacity.

C. Reliability and Capacity Modeling

Based on the architecture, a Reliability Block Diagram (RBD) is constructed (Fig. 3), incorporating:

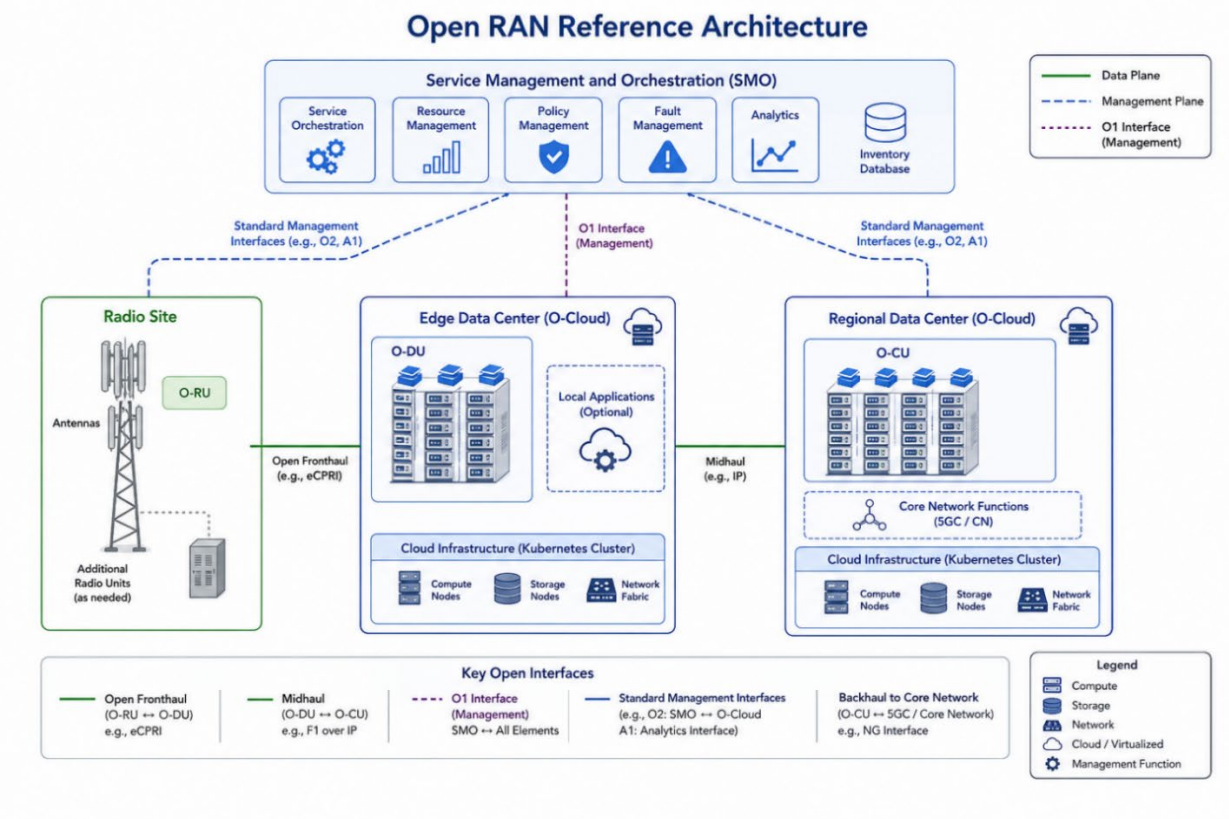

Figure 2. Open RAN Architecture – Functional View

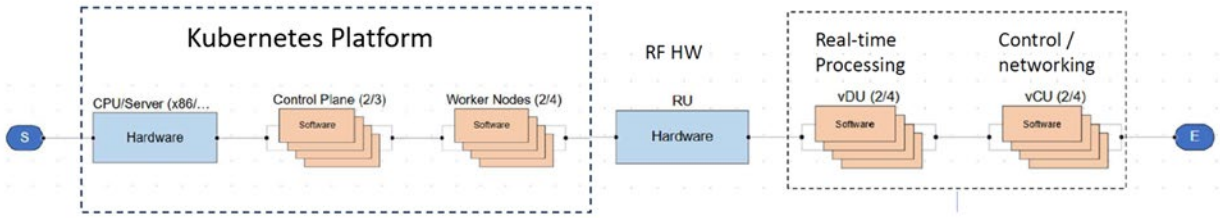

Figure 3. Reliability block diagram for an Open RAN system with a Kubernetes infrastructure

- Functional components (RU, vDU, vCU)
- Infrastructure components (Kubernetes platform)

Unlike conventional RBD models, this approach introduces effective capacity to capture realistic system behavior.

III. Mathematical Model

A. Model Assumptions

The system consists of:

- Kubernetes infrastructure
- RF hardware (RU)
- Software functions (vDU, vCU)

vDU and vCU operate on Kubernetes and are constrained by infrastructure capacity.

B. Input Parameters

Key parameters include:

- Reliability parameters: failure rate λ_f, repair rate μ_f, availability A_f
- Traffic load: λ
- Capacity parameters: processing rates μ
- Redundancy parameters: k-of-n structure
- Overhead factors: virtualization overhead α

C. Structural Reliability Model

System availability is defined using k-out-of-n conditions:

$$R_{k,n} = \sum_{i=k}^{n} \binom{n}{i} A^i (1-A)^{n-i} \tag{1}$$

These conditions determine whether the system is functionally operational.

D. Capacity Model with Kubernetes Constraints

1) Infrastructure Capacity

$$C_{K8s} = f(\text{number of worker nodes}) \tag{2}$$

2) Effective Capacity of vDU and vCU

$$C_x = \min(C_x^{node}, C_{K8s}) \tag{3}$$

3) Virtualization Overhead

$$C_x^{eff} = C_x^{node}(1 - \alpha_x) \tag{4}$$

$$C_x = \min(C_x^{node}(1 - \alpha_x), C_{K8s}) \tag{5}$$

E. Utilization and System Stability

$$\rho_x = \frac{\lambda}{C_x} \tag{6}$$

$$\rho_{system} = \max_{x \in \{vDU, vCU, K8s\}} \rho_x \tag{7}$$

In this model, the traffic parameter λ is treated as a normalized processing demand rate, rather than a fixed physical unit such as Gbps or session count. Specifically, λ represents the workload imposed on the system relative to its processing capacity μ, and therefore depends on the system configuration, including the number of nodes, processing rates, and virtualization overhead.

For example, the value “204” in Fig. 4 does not correspond to a specific bandwidth, but to a relative load level at which system utilization approaches the stability limit under the given configuration.

In practical deployments, this normalized traffic can be calibrated to real-world units (e.g., Gbps or session counts) using measured traffic data and system capacity.

F. Overload Probability

$$P_{overload} = P(\rho_{system} \geq 1) \tag{8}$$

G. Stability Classification

- $\rho \leq 0.8$: Safe
- $0.8 < \rho \leq 0.9$: Warning
- $0.9 < \rho < 1$: Critical
- $\rho \geq 1$: Overload

The normalization of traffic and the resulting ability to interpret system behavior across configurations further enhance the model’s applicability to real-world system design and operational analysis.

IV. Sensitivity Analysis

Sensitivity analysis is conducted with virtualization overhead as a key parameter.

As shown in Fig. 4:

- vCU becomes the bottleneck
- System utilization follows the maximum component utilization
- At $\lambda \approx 204$, utilization reaches 1
- Overload probability increases sharply (tipping point)

This confirms that system instability occurs **abruptly rather than gradually**.

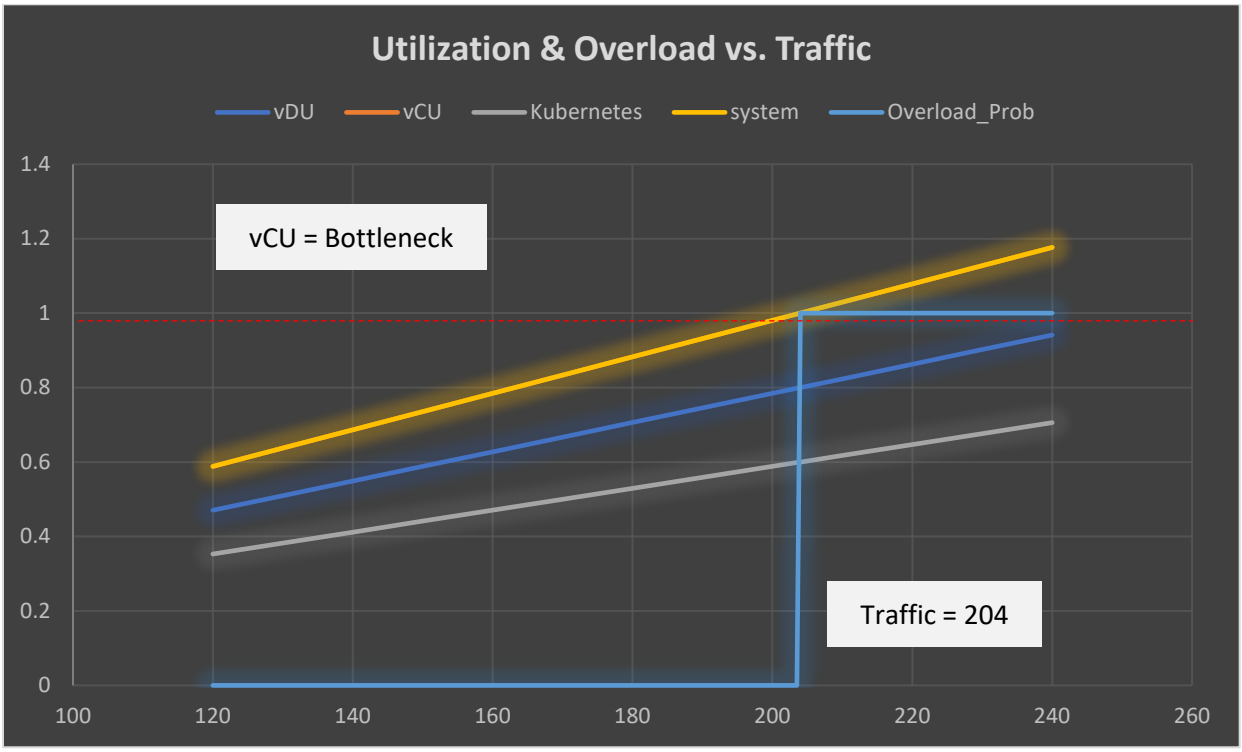

Figure 4. Utilization and overload probability as normalized traffic (λ) increases. The value "204" represents a configuration-dependent load level at which the system approaches instability.

V. Discussion

The model reveals:

- Kubernetes acts as a capacity ceiling
- vDU and vCU performance depends on the infrastructure
- The most constrained layer determines system stability

This provides a realistic view of system behavior under dynamic load conditions.

VI. Conclusion

This paper demonstrates that the stability of Open RAN systems is governed not only by structural reliability but also by capacity constraints and traffic load.

The proposed model enables:

- Quantitative system stability evaluation
- Bottleneck identification
- Predictive reliability analysis

In particular, the model provides a quantitative basis for addressing practical system-level questions, including:

• Identification of bottleneck components under dynamic load conditions

• Evaluation of infrastructure-induced performance constraints

• Explanation of instability occurring despite redundancy

• Bridging the gap between test environments and real-world operation

These capabilities enable more informed system design and operational decision-making.

Acknowledgment

The author would like to thank Toshio Takahashi for valuable discussions and insights.

References

[1] O-RAN Alliance, *O-RAN Architecture Description*, 2020.
[2] O-RAN Alliance, *O-RAN: Towards an Open and Smart RAN*, White Paper, 2018.
[3] M. Polese et al., “Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges,” *IEEE Communications Surveys & Tutorials*, 2022.
[4] Cloud Native Computing Foundation (CNCF), *Cloud Native Definition v1.0*, 2018.
[5] ETSI, *Network Functions Virtualization (NFV) Architectural Framework*, ETSI GS NFV 002.
[6] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, Wiley, 2002.
[7] L. Kleinrock, *Queueing Systems, Volume 1*, Wiley, 1975.

[8] J. D. Musa, A. Iannino, and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*, McGraw-Hill, 1987.

