




Performance Evaluation of Call Admission Control Strategy in Cloud Radio Access Network using Formal Methods

Maroua Idi^{1,2}^a, Sana Younes^{1,2}^b and Riadh Robbana^{1,3}^c

¹LIPSIC Laboratory, Faculty of Sciences of Tunis, University of Tunis El Manar, Tunis 2092, Tunisia

²Faculty of Sciences of Tunis, University of Tunis El Manar, Tunis 2092, Tunisia

³National Institute of Applied Sciences and Technology, University of Carthage, Tunis 1080, Tunisia

Keywords: Cloud Radio Access Network, Virtual Machine, Hysteresis, Stochastic Model Checking, CTMC, PRISM.

Abstract: For the fifth-generation (5G), Cloud Radio Access Network (C-RAN) has been proposed as a cloud architecture to provide a common connected resource pool management. In this regard, considering the rapidly changing in network traffic load, the efficient management of radio resources is a challenge.


Call Admission Control (CAC) is a resource allocation mechanism to guarantee the Quality of Service (QoS) to User Equipment (UE) in a mobile cellular network. This paper proposes a new CAC schema, based on a hysteresis mechanism, named Virtual Machine Hysteresis Allocation Strategy (VMHAS) in the context of C-RAN. We aim to provide a good QoS by improving the blocking probability of calls, adjusting the amount of active VMs being provisioned for the current traffic load, and providing a load balancing in the considered C-RAN. We use probabilistic model checking to evaluate the performance of the proposed strategy. First, we model the VMHAS CAC schema with Continuous-Time Markov Chains (CTMCs). Then, we specify QoS requirements through the CTMC using the Continuous-time Stochastic Logic (CSL). Finally, we quantify the performance measures of the considered strategy by checking CSL steady-state and transient formulas using the PRISM model checker.


1 INTRODUCTION


Cloud Radio Access Network (C-RAN) is a novel mobile network architecture based on a virtualization technology that has emerged as a promising architecture to efficiently address the challenges of the fifth-generation (5G) cellular networks, such as spectrum efficiency and energy reduction. The concept was first proposed by IBM in (Lin et al., 2010) with the name of wireless network cloud to reduce networking costs and achieve more flexible network capabilities. Then this concept was described in detail by China Mobile Research Institute in 2011 (Chen and Duan, 2011). In contrast to the traditional access networks, the main idea behind C-RAN (Checko et al., 2014) is to decompose the traditional Base Stations (BS) into Base Band Units (BBUs) and Remote Radio Heads (RRHs) that are respectively responsible for baseband and radio functionalities. Although the

RRHs are distributed and deployed with an antenna at the cell site, the BBUs are grouped in a data center called the BBU pool. The connection between the BBUs and RRHs is referred to as the fronthaul links and is done via optical transport network. In this architecture, all BBUs functions could be implemented on standard hardware and executed on Virtual Machines (VMs) (Bhamare et al., 2018). Hence, they could serve each User Equipment (UE) by generating a VM to provide computing resources as common data centers do in a cloud-based system (Urgaonkar et al., 2010). In the BBU pool, each UE has its own corresponding VM (Wang et al., 2017), (Haberland et al., 2013). The number of VMs generated by each BBU is restricted due to the limited processing ability of each BBU. (Chen and Duan, 2011), which means that each BBU can only support a certain number of UEs.

Radio resource allocation always remains a big challenge for all cellular networks; even for the C-RAN, it is crucial to increase spectral efficiency while guaranteeing a good Quality of Service (QoS). Therefore, Call Admission Control (CAC) is one of the fun-

^a <https://orcid.org/0000-0002-6467-8887>

^b <https://orcid.org/0000-0002-4883-3381>

^c <https://orcid.org/0000-0001-5736-4137>

damental strategies for radio resource management that decides to accept or reject a new UE connection demand based on the current cell load, the QoS of the new UE demand, and of the ongoing traffic.

In existing works (Sigwele et al., 2014), (Sigwele et al., 2015), all VMs of BBUs are always activated, waiting for the arrival of calls in the case of high or low traffic load. Whereas when a VM is activated, it consumes energy. In fact, an idle VM, which is not busy by call, consumes 60 to 80 percent of the energy consumed by an occupied VM (Duan et al., 2015). In this context, in order to preserve energy, a common method makes non-occupying BBUs in sleep mode (Sigwele et al., 2017a).

In our work, we use the sleep mode in the developed strategy but differently. In fact, we put VMs in sleep mode, level by level in all BBUs, using the hysteresis mechanism. Moreover, let us note that it is essential to take care of balancing the load between the different BBUs through an appropriate VM allocation strategy because the excessive or insufficient resource utilization in the BBU impacts the virtual BBU's performance and the physical equipment's maintenance costs.

In this regard, we propose a new CAC scheme, called Virtual Machine Hysteresis Allocation Strategy (VMHAS). The proposed VMHAS, on the one hand, adjusts the number of active VMs in BBUs by making not using VMs in sleep mode and ensuring a low call blocking probability. On the other hand, it guarantees the load balancing between BBUs. To achieve this, we propose using the hysteresis mechanism based on the division of resources (VMs) to levels. Each level will be activated when the used resource attains an activated threshold. Similarly, the deactivation of level is done when the using resources are less than a deactivation threshold. Let us note that in the hysteresis mechanism, the deactivation threshold is always strictly less than the activation threshold because the fluctuations of the reserved resources from levels (activation and deactivation of VMs) should be the minimum possible as the rapid switching between levels has costs.

By applying the hysteresis mechanism in our VM strategy, we propose to divide all BBUs into three levels of VMs. In the first level of all BBUs, VMs are always active and in idle mode, waiting for the arrival of calls. However, in the remaining two levels of all BBUs, VMs are deactivated and in sleep mode and will be activated in need. In order to achieve a load balancing between all BBUs, we propose to assign, level by level, the UE calls by available VMs in all BBUs, and we put the call to the least loaded BBU. The second (resp. third) level of VMs will be

activated simultaneously in all BBUs when the number of occupied VMs attains the first (resp. second) hysteresis activating threshold. When the current traffic load in all BBUs decreases, we aim to reduce the number of idle VMs. Hence, the third (resp. second) level of VMs will be deactivated, simultaneously in all BBUs, when the number of occupied VMs decreases and is lower than the descending thresholds. Recall that we choose deactivation thresholds that are lower than activation thresholds to reduce the switching operation between levels.

In this paper, we use Probabilistic Model Checking (PMC) to analyze the performance of the proposed VMHAS. PMC is a probabilistic extension of the model checking formal verification technique, used to analyze stochastic systems in different domains (Kwiatkowska et al., 2005). It requires two inputs: a description of the system and a specification of requirements under the system expressed in temporal logics. In this work, we develop a Continuous-Time Markov Chains (CTMC) model to describe our CAC schema. We specify QoS requirements in terms of diminishing blocking probability, ensuring the load balancing between BBUs, and adjusting the number of active VMs being provisioned for the current traffic load. These requirements are expressed by checking CSL steady-state and transient formulas of the system using the PRISM model checker (Kwiatkowska et al., 2011) to perform the VMHAS strategy.

The rest of the paper is organized as follows. In section 2, we discuss the related work of CAC schemes in the context of C-RAN and the use of the hysteresis mechanism. In section 3, we give a brief description of CTMC and CSL. Then, in section 4, a performance model of the considered VMHAS is presented. In section 5, we present and discuss the results of formal verification of QoS properties. Finally, we conclude the paper.

2 RELATED WORK

In this section, we enumerate some works that present CAC algorithms in the context of C-RAN. Then, we focus on works that use the hysteresis mechanism in order to ensure optimum reservation and utilization of resources.

CAC is a mechanism that can play a key role in providing guaranteed QoS and avoiding traffic congestion in all cellular networks. In this context, previous works have been proposed in (Younes and Benbarek, 2017), (Younes and Idi, 2018) to treat CAC schemes for the fourth generation, where the BS is not shared.

In the context of C-RAN, Sigwele et al. proposed in (Sigwele et al., 2014) an algorithm for CAC to ensure the QoS needs of the requested call. This algorithm collects and uses traffic information to verify the existence of sufficient resources, and it assigns the incoming call to the less-load BBU in the cluster of BBUs. When all BBUs are saturated, and the QoS requirements are violated, the incoming calls are blocked. To solve this problem, the authors take the benefit of cloud elasticity to increase the processing capacity of BBUs. In (Khan et al., 2015), the authors proposed a self-organized C-RAN. The proposed network architecture is formulated as an optimization problem and can balance network traffic by reducing the number of blocked calls and improving the QoS.

In (Sigwele et al., 2015), (Sigwele et al., 2017b) and (Al-Maitah et al., 2018) authors proposed a CAC schemes using Fuzzy Logic for heterogeneous traffic classes. They evaluate with simulation the call blocking probabilities.

To ensure efficient utilization of BBUs, the idea of the model presented in (Sigwele et al., 2017a) is to act with a fixed amount of BBUs and, according to the demand, to deactivate the idle BBUs and reactivate them only in case of overloading. In [(Gakhar et al., 2006), (Levy et al., 2004), (Halberstadt et al., 1995)], authors proposed CAC schemes based on hysteresis mechanism in old cellular access networks. The allocation mechanism in (Gakhar et al., 2006) was proposed for traffic in IEEE 802.16 broadband wireless network, and it dynamically modified the number of resources reserved between a minimum and maximum number depending on the number of active connections. It considers one, two, and multiple thresholds in the three cases studied. In (Levy et al., 2004), the authors consider multiple thresholds. Hence, the reserved bandwidth varies from one to another threshold until the number of channels being used reaches a prefixed threshold. A mechanism to optimize resources allocations in ATM networks was proposed in (Halberstadt et al., 1995). In this model, the passage of the bandwidth to a superior or inferior reservation is attained as a function of the state of the client queue at the ATM switch.

3 PRELIMINARIES

This section introduces the basic concepts of formalisms that we use to evaluate performance measures for the studied CAC scheme. We start by presenting labelled CTMC, and then we recall the logic CSL. For more details we refer to (Kulkarni, 2016) for CTMC and to (Aziz et al., 2000) for CSL.

3.1 CTMC

A labelled CTMC \mathcal{M} is a tuple (S, \mathbf{R}, L) where S is a finite set of states, $\mathbf{R} : S \times S \rightarrow \mathcal{R}^+$ is the rate matrix and $L : S \rightarrow 2^{AP}$ is the labelling function which assigns to each state $s \in S$, the set $L(s)$ of atomic propositions $a \in AP$ that are valid in s . The finite set of atomic propositions is denoted by AP . \mathbf{Q} , the infinitesimal generator, can be deduced as $\mathbf{Q}(s, s') = \mathbf{R}(s, s')$ if $s \neq s'$ and $\mathbf{Q}(s, s) = -\sum_{s' \in S} \mathbf{R}(s, s')$.

A Path. Through a CTMC, a path is an alternating sequence $\sigma = s_0 t_0 s_1 t_1 \dots$ with $\mathbf{R}(s_i, s_{i+1}) > 0$ and $t_i \in \mathcal{R}^+$ for all $i \geq 0$. t_i defines the amount of time spent in state s_i . Let's call by $path_s$ the set of paths through \mathcal{M} starting from the state s .

State Probabilities. There are two types of state probabilities in a CTMC: transient probabilities consider the system at a time t and steady-state probabilities when the system reaches an equilibrium if it exists. Let us denote by $\Pi_s^{\mathcal{M}}(t)$ the transient distribution at time t of \mathcal{M} starting at $t = 0$ from the initial state s . The probability to be in state s' at time t starting initially from s will be denoted by $\Pi_s^{\mathcal{M}}(s', t)$. The steady-state probability to be in state s' is $\Pi_s^{\mathcal{M}}(s') = \lim_{t \rightarrow \infty} \Pi_s^{\mathcal{M}}(s', t)$. If \mathcal{M} is ergodic (irreducible), $\Pi_s^{\mathcal{M}}(s')$ exists and it is independent of the initial distribution that we will denote by $\Pi^{\mathcal{M}}(s')$. Also, we denote by $\Pi^{\mathcal{M}}$ the steady-state probability vector. For $S' \subseteq S$, we denote by $\Pi_s^{\mathcal{M}}(S', t)$ (resp. $\Pi^{\mathcal{M}}(S')$) the transient probability to be in states of S' , $\Pi_s^{\mathcal{M}}(S', t) = \sum_{s' \in S'} \Pi_s^{\mathcal{M}}(s', t)$ (the steady-state probability to be in states of S' , $\Pi^{\mathcal{M}}(S') = \sum_{s' \in S'} \Pi^{\mathcal{M}}(s')$).

3.2 CSL

This subsection presents CSL, which allows specifying properties over CTMCs. CSL is an extension of CTL (Computational Tree Logic) (Clarke et al., 1986) with two probabilistic operators that refer to steady-state and transient behaviors of the underlying system.

Let p be a probability threshold, \triangleleft be a comparison operator with $\triangleleft \in \{\leq, \geq, <, >\}$, and I be an interval of real numbers. The set of states that satisfy ϕ property is denoted by S_ϕ , and the satisfaction relation is denoted by \models .

The syntax and semantic of CSL are defined

$$\begin{array}{ll} s \models true & \text{for all } s \in S \\ s \models a & \text{iff } a \in L(s) \\ \text{by: } s \models \neg\phi & \text{iff } s \not\models \phi \\ s \models \mathcal{P}_{\triangleleft p}(\phi_1 \ \mathcal{U}^I \phi_2) & \text{iff } Prob^{\mathcal{M}}(s, \phi_1 \ \mathcal{U}^I \phi_2) \triangleleft p \\ s \models S_{\triangleleft p}(\phi) & \text{iff } \Pi_s^{\mathcal{M}}(S_\phi) \triangleleft p \end{array}$$

In this paper, we will use the probabilistic operators $\mathcal{P}_{\triangleleft p}(\phi_1 \mathcal{U}^I \phi_2)$ and $\mathcal{S}_{\triangleleft p}(\phi)$ to define and quantify performance measures of the studied system. In fact, these operators are referring to transient and steady state behavior of the considered system.

The operator $\mathcal{P}_{\triangleleft p}(\phi_1 \mathcal{U}^I \phi_2)$ asserts that the probability measure of paths satisfying $\phi_1 \mathcal{U}^I \phi_2$ meets the bound given by $\triangleleft p$. Whereas, the path formula $\phi_1 \mathcal{U}^I \phi_2$ asserts that ϕ_2 will be satisfied at some time $t \in I$ and that at all preceding time ϕ_1 holds. $Prob^{\mathcal{M}}(s, \phi_1 \mathcal{U}^I \phi_2)$ denotes the probability measure of all paths σ starting from s ($\sigma \in paths_s$) satisfying $\phi_1 \mathcal{U}^I \phi_2$ i.e. $Prob^{\mathcal{M}}(s, \phi_1 \mathcal{U}^I \phi_2) = Prob\{\sigma \in paths_s \mid \sigma \models \phi_1 \mathcal{U}^I \phi_2\}$.

Recall that the verification of time bounded until formula $\mathcal{P}_{\triangleleft p}(\phi_1 \mathcal{U}^I \phi_2)$ for a CTMC \mathcal{M} requires the computation of $Prob^{\mathcal{M}}(s, \phi_1 \mathcal{U}^I \phi_2)$. This measure can be computed by transient analysis of another CTMC \mathcal{M}' which is derived from \mathcal{M} . Let $\mathcal{M}[\phi]$ be the CTMC defined from $\mathcal{M} = (S, \mathbf{R}, L)$, by making all ϕ -states in \mathcal{M} absorbing, i.e. $\mathcal{M}' = (S, \mathbf{R}', L)$ where $\mathbf{R}'(s, s') = \mathbf{R}(s, s')$ if $s \not\models \phi$ and 0 otherwise.

In this paper, we will use the formula $\mathcal{P}_{\triangleleft p}(true \mathcal{U}^{[t,t]} \phi_2)$ which is a particular case of ($\phi_1 = true$) and in a specific time t . In this particular case $\mathcal{M}' = \mathcal{M}$, $\phi_1 \wedge \phi_2 = \phi_2$ and the verification of this formula requires the computation of transient distribution at time t of the considered model \mathcal{M} without doing any transformation.

$$s \models \mathcal{P}_{\triangleleft p}(true \mathcal{U}^{[t,t]} \phi_2) \text{ iff } \Pi_s^{\mathcal{M}}(S_{\phi_2}, t) = \sum_{s' \models \phi_2} \Pi_s^{\mathcal{M}}(s', t) \triangleleft p \quad (1)$$

The operator $\mathcal{S}_{\triangleleft p}(\phi)$ asserts that the steady-state probability for ϕ -states meets the bound $\triangleleft p$. The verification of the steady-state operator requires the computation of the steady-state probability to be in ϕ -states.

$$s \models \mathcal{S}_{\triangleleft p}(\phi) \text{ iff } \Pi_s^{\mathcal{M}}(S_{\phi}) = \sum_{s' \models \phi} \Pi_s^{\mathcal{M}}(s') \triangleleft p \quad (2)$$

In this work, we will also use two reward operators from Continuous Stochastic Reward Logic (CSRL). The CSRL (Haverkort et al., 2002) is an extension of CSL by adding constraints over rewards. $\mathcal{E}_J(\phi)$, the steady-state reward operator, asserts that the expected reward rate for ϕ -states lies in J (J is an interval of real numbers). The transient operator reward $\mathcal{E}_J^t(\phi)$ asserts that the expected instantaneous reward rate at time t for ϕ -states lies in J .

Let $\rho : S \rightarrow \mathcal{R}^+$ be a *reward structure* that assigns to each state $s \in S$ a reward value $\rho(s)$. The verification of these reward formulas $\mathcal{E}_J(\phi)$ (resp. $\mathcal{E}_J^t(\phi)$) requires the computation of the steady-state

(resp. transient at t) distribution $\Pi_s^{\mathcal{M}}$ of the considered \mathcal{M} .

$$\begin{aligned} s \models \mathcal{E}_J(\phi) & \text{ iff } \sum_{s' \in S_{\phi}} \Pi_s^{\mathcal{M}}(s') \cdot \rho(s') \in J \\ s \models \mathcal{E}_J^t(\phi) & \text{ iff } \sum_{s' \in S_{\phi}} \Pi_s^{\mathcal{M}}(s', t) \cdot \rho(s') \in J \end{aligned} \quad (3)$$

4 SYSTEM DESCRIPTION AND FORMAL MODEL OF VMHAS

In this section, we first describe the C-RAN architecture adopted in this paper. Then, we present the VM Hysteresis Allocation Strategy (VMHAS) that we propose. After that, we develop the algorithmic description of VMHAS. Finally, we give the Markovian model of the proposed VMHAS.

4.1 System Description

The C-RAN architecture that we consider in this paper, as shown in Fig. 1, is composed of three main components: a centralized BBU pool on the cloud that contains a number of BBUs, a cell with a number of distributed RRHs, and the fronthaul links used to transmit baseband signals between the BBU pool and RRHs. Given the adopted C-RAN architecture, each BBU on the BBU pool is composed of a set of VMs, and it can support one or more RRHs. We assume that all BBUs are identical, they have the same number of VMs, and a VM can only serve one UE.

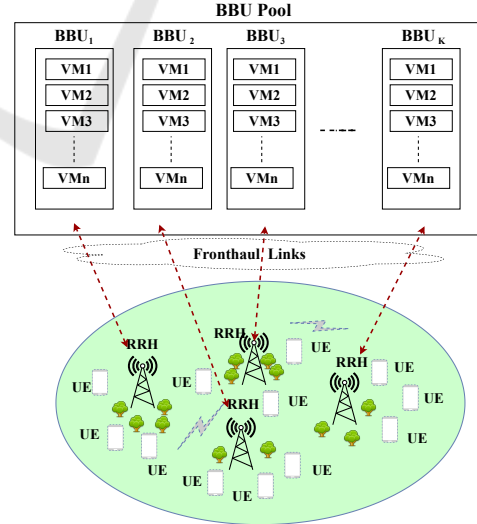


Figure 1: C-RAN considered architecture.

4.2 Proposed VM Hysteresis Allocation Strategy (VMHAS)

This subsection explains in detail the strategy that we propose. Recall that we aim to achieve maximum accepted calls while adjusting the number of active VMs in BBUs needed to serve all UE demands and ensure a load balancing between BBUs.

In the proposed VMHAS, each BBU is divided into three levels of VMs, as shown in Fig. 2. The two hysteresis activating thresholds are V_{m_1} and V_{m_2} , while the two hysteresis deactivating thresholds are T_1 and T_2 with $T_1 < V_{m_1}$, and $T_2 < V_{m_2}$. A VM can be in an active or sleep mode, as shown in Fig. 3. When a VM is activated, it can be in two different modes: idle or busy. The idle mode is when the VM is activated but not occupied by a call. Whereas the busy mode is when the VM is activated and occupied by a call.

Initially, for all BBUs, all VMs in the first level, containing V_{m_1} VMs, are always activated and in idle mode, while the remaining VMs belonging to the other two levels are deactivated and in sleep mode. When a call arrives, it will be assigned to the available VM in the least load BBU. By default, when the current traffic load is the same in all BBUs, the first BBU will serve the incoming call.

The second level, containing $(V_{m_2} - V_{m_1})$ VMs, will be activated when the first level in all BBUs reaches the maximum capacity V_{m_1} . Therefore, when a call arrives, and it is assigned to the last available VM in the first level, all VMs in the second level will be activated, simultaneously, in all BBUs and be in idle mode waiting for the arrival of calls.

Similarly, the third level, containing $(V_{max} - V_{m_2})$ VMs, will be activated when the second level reaches the maximum capacity V_{m_2} in all BBUs. From the description above, it is clear that the activation of the second (resp. third) level of VMs in all BBUs depends on traffic demands and the current traffic load in the system. Therefore, when the current traffic load decreases and in order to diminish the number of VMs in idle mode, we adopt to deactivate level by putting VMs to sleep mode. Note that we talk about VMs deactivation when the system is in the second or the third level. When the second (resp. third) level is activated, and the current number of busy VMs in the system is strictly lower than T_1 (resp. T_2) in all BBUs, then the corresponding VMs of the second (resp. third) level will be deactivated and be in sleep mode.

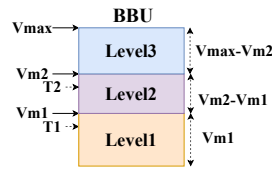


Figure 2: BBU with three levels of VMs.

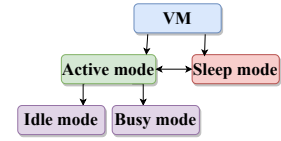


Figure 3: VM-modes in a BBU.

4.3 Proposed Algorithm of VMHAS

In this subsection, we present the algorithmic description of VMHAS in the case general of K-BBUs. We use a Markovian process to model our strategy in which the arrival of calls follows a Poisson process and the service time (duration of the call) follows an exponential distribution. Therefore, under these Markovian hypotheses, the arrival of different calls and the departure of ongoing calls cannot trigger simultaneously.

Algorithm 1: Proposed algorithm of VMHAS in the case of K-BBUs.

```

K: Total number of BBUs;
Vmax: Max number of VMs in a BBU;
Vm2: Hysteresis Level 3 activating threshold;
Vm1: Hysteresis Level 2 activating threshold;
T2: Hysteresis Level 3 deactivating threshold;
T1: Hysteresis Level 2 deactivating threshold;
Vk: Number of VMs in busy mode in the kth
    BBU where (1 ≤ k ≤ K);
/* Number of active VMs is initialized
   to Vm1. */
L = Vm1;
/* All BBUs are empty. */
for (1 ≤ k ≤ K) do
    Vk = 0
ActiveLevel2 ← False;
ActiveLevel3 ← False;
/* Two methods used to verify the
   arrival (resp. the departure) of a
   call. */
CallA = CallArrived();
CallD = CallDeparture();
While (CallA or CallD) do
    if CallA then
        /* Put the call in the first BBU.
           */
        if (∀ 1 ≤ k ≤ K, V1 = V2... = VK) then
            V1 = V1 + 1;
        /* Put the call in the least
           loaded BBU. */
        else if (∀ 1 ≤ k ≤ K, Vk < L) then
            Vl = min Vk;
            Vl = Vl + 1;
    
```

Algorithm 1: Proposed algorithm of VMHAS in the case of K-BBUs (cont.).

```

/* For each BBU, activate the
second level of VMs. */
if not ActiveLevel2 then
    if ( $\forall 1 \leq k \leq K, V_k = V_{m_1}$ ) then
         $L = V_{m_2}$ ;
        ActiveLevel2 = True;
/* For each BBU, active the third
level of VMs. */
if not ActiveLevel3 then
    if ( $\forall 1 \leq k \leq K, V_k = V_{m_2}$ ) then
         $L = V_{max}$ ;
        ActiveLevel3 = True;
/* Reject the call because all
VMs in all BBUs are occupied.
*/
if ( $\forall 1 \leq k \leq K, V_k = V_{max}$ ) then
    reject call ;
if CallD then
    /* Function returning a BBU from
    which a call departed. */
     $k = departedcall()$ ;
     $V_k = V_k - 1$ ;
    /* For each BBU, deactivate the
    second level of VMs */
    if ActiveLevel2=True then
        if ( $\forall 1 \leq k \leq K, V_k < T_1$ ) then
             $L = V_{m_1}$ ;
            ActiveLevel2 = False;
    /* For each BBU, deactivate the
    third level of VMs */
    if ActiveLevel3=True then
        if ( $\forall 1 \leq k \leq K, V_k < T_2$ ) then
             $L = V_{m_2}$ ;
            ActiveLevel3 = False;
    CallA = CallArrived();
    CallD = CallDeparture();
    
```

4.4 Model Analysis for 2-BBUs

The use of discrete-state approaches to model the performance of large-scale systems is fundamentally prevented by the state-space explosion problem, which causes an exponential increase of the reachable state space as a function of the number of components which constitute the model. This is suitable for our model, which is modeled by multidimensional CTMC and consists of large numbers of components (each dimension represents the number of active VMs in a BBU). Therefore, in order to represent our VMHAS model, we choose to discuss a labelled Markov model

in a special case of two BBUs ($K = 2$)

Let us remember that we assume that the arrival processes of traffic are independent and follow a Poisson distribution with a rate equal to λ . We suppose that the holding time of VMs is exponentially distributed with a mean $1/\mu$.

Based on these assumptions for arrival and service rates, the proposed VMHAS can be modeled by a multidimensional homogeneous CTMC \mathcal{M} , presented in Fig. 4.

Obviously, the obtained \mathcal{M} is composed of three blocks relative to the number of levels in the hysteresis mechanism. The first (resp. second) contains states of S_1 (resp. S_2) (see Eq. 4 and Eq. 5) relative to the activation of the second (resp. third) level of VMs. The third block contains states of S_3 (see Eq. 6) where VMs are activated in all BBUs.

It is easily seen that the two transitions with continuous lines (marked in blue) represent the activation for the second and the third level of VMs, while the transitions with broken lines (marked in green) represent the deactivation for the second and the third level of VMs.

The state space is given by:

$$S = S_1 \cup S_2 \cup S_3$$

In state (i, j, l) , i (resp. j) represents the number of busy VMs in BBU₁ (resp. BBU₂), and l represents the activate level of VMs (1, 2 or 3):

$$S_1 = \{(i, j, 1); 1 \leq i \leq V_{m_1} \text{ and } 0 \leq j \leq i - 1\} \cup \{(i, j, 1); 0 \leq j \leq V_{m_1} - 1 \text{ and } 0 \leq i \leq j\} \quad (4)$$

$$S_2 = \{(i, j, 2); T_1 \leq i \leq V_{m_2} \text{ and } 0 \leq j \leq i - 1\} \cup \{(i, j, 2); T_1 \leq j \leq V_{m_2} - 1 \text{ and } 0 \leq i \leq j\} \quad (5)$$

$$S_3 = \{(i, j, 3); T_2 \leq i \leq V_{max} \text{ and } 0 \leq j \leq i - 1\} \cup \{(i, j, 3); T_2 \leq j \leq V_{max} \text{ and } 0 \leq i \leq j\} \quad (6)$$

By the particular structure of the obtained \mathcal{M} , we can calculate the number of states in S_1 , S_2 and S_3 which is denoted respectively by N_1 , N_2 , and N_3 :

$$N_1 = \sum_{i=1}^{V_{m_1}} i + \sum_{i=0}^{V_{m_1}-1} (i+1) = 2 \sum_{i=1}^{V_{m_1}} i$$

$$N_2 = \sum_{i=T_1}^{V_{m_2}} i + \sum_{i=T_1}^{V_{m_2}-1} (i+1) = 2 \sum_{i=T_1+1}^{V_{m_2}} i + T_1$$

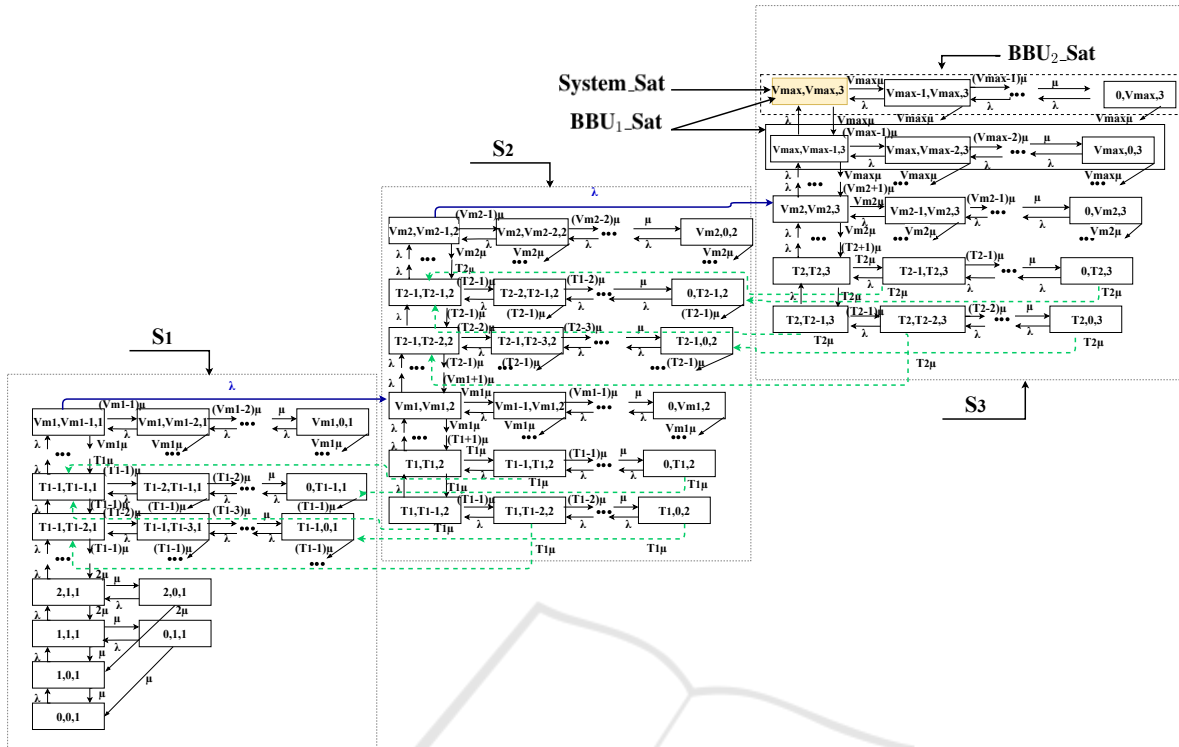


Figure 4: CTMC of the proposed VMHAS.

$$N_3 = \sum_{i=T_2}^{V_{max}} i + \sum_{i=T_2}^{V_{max}} (i+1) = 2 \sum_{i=T_2}^{V_{max}} i + V_{max} - T_2 + 1$$

The number of states N of the obtained \mathcal{M} can be deduced by adding N_1 , N_2 and N_3 . It is given by Eq. 7.

$$N = V_{m1}^2 + V_{m2}^2 + (V_{max} + 1)^2 + V_{m1} + V_{m2} - T_1^2 - T_2^2 \quad (7)$$

Similarly, the number of the transitions can be obtained by the Eq. 8.

$$TN = 3(V_{max}^2 + V_{m2}^2 + V_{m1}^2) + 4V_{max} + V_{m2} + V_{m1} - 3(T_1^2 + T_2^2) + 2(T_1 + T_2) - 2 \quad (8)$$

5 NUMERICAL RESULTS

In this section, we present numerical results of the performance evaluation of the proposed VMHAS. These results are obtained by verifying CSL formulas under the VMHAS model that we proposed in the subsection 4.2. Therefore, to check CSL formulas that specify performances requirements, we label the states of \mathcal{M} , presented in Fig. 4, with atomic propositions which characterize the states. Let us consider the following set of atomic propositions AP.

$$AP = \{\text{System_Sat}, \text{BBU}_1_Sat, \text{BBU}_2_Sat\}$$

The atomic proposition System_Sat is assigned to states in which the call is blocked in the system. BBU₁_Sat (resp. BBU₂_Sat) is assigned to states in which the call is blocked in BBU₁ (resp. BBU₂).

The obtained satisfaction sets are marked in Fig. 4 and defined formally by:

$$\begin{aligned} S_{\text{System_Sat}} &= \{(i, j, l) \mid i = V_{max} \ \& \ j = V_{max} \ \& \ l = 3\} \\ S_{\text{BBU}_1_Sat} &= \{(i, j, l) \mid i = V_{max} \ \& \ 0 \leq j \leq V_{max} \ \& \ l = 3\} \\ S_{\text{BBU}_2_Sat} &= \{(i, j, l) \mid 0 \leq i \leq V_{max} \ \& \ j = V_{max} \ \& \ l = 3\} \end{aligned}$$

Obviously:

$$S_{\text{System_Sat}} = S_{\text{BBU}_1_Sat} \cap S_{\text{BBU}_2_Sat}.$$

In order to construct and solve \mathcal{M} , we use the probabilistic model checker PRISM (Kwiatkowska et al., 2011). This tool is a high-level modelling language, and formulas are checked automatically. Numerical results that we present in this section are obtained with the parameters presented in Table 1.

The choice of the value of thresholds ($T_1 < V_{m1}$ and $T_2 < V_{m2}$) is justifiable because we need some reserve VMs to avoid the rapid activating or deactivating of the second (resp. third) level.

Table 1: Experimental Parameters.

Parameters	Value
K :Total number of BBUs	2
$Vmax$:Max number of VMs in a BBU	100
Vm_2 :Hysteresis Level 3 activating threshold	60
Vm_1 :Hysteresis Level 2 activating threshold	40
T_2 : Hysteresis Level 3 deactivating threshold	56
T_1 :Hysteresis Level 2 deactivating threshold	36
$1/\mu$: The mean VM holding time (per minute)	1

The size of the obtained \mathcal{M} (states and transitions number) is calculated by PRISM and is equal to 11069 and 32986. These results are equal to the mathematics formulas established by Eq. 7 and Eq. 8.

Now, we present the performance evaluation of VMHAS obtained by checking steady-state and transient formulas.

5.1 Checking Steady-state Formulas

The verification of steady-state formulas needs the computation of steady-state distribution $\Pi^{\mathcal{M}}$ of the considered \mathcal{M} . It is clear that the obtained \mathcal{M} of VMHAS presented in subsection 4.4 is ergodic (irreducible), so the steady-state probability vector $\Pi^{\mathcal{M}}$ of \mathcal{M} exists and is unique and it is independent of the initial distribution.

5.1.1 $\mathcal{S}_{=?}(\phi)$

The verification of this formula is presented by Eq. 2. In order to compute the steady-state call blocking probability in the system, in BBU₁ and in BBU₂, we check the following formulas:

- $\mathcal{S}_{=?}(\text{System.Sat})$: specifies the steady-state call blocking probability for two BBUs (system). This measure is equal to $\Pi^{\mathcal{M}}(S_{\text{System-Sat}})$.
- $\mathcal{S}_{=?}(\text{BBU}_1.\text{Sat})$ (resp. $\mathcal{S}_{=?}(\text{BBU}_2.\text{Sat})$): specifies the steady-state call blocking probability for BBU₁ (resp. BBU₂). This measure is equal to $\Pi^{\mathcal{M}}(S_{\text{BBU}_1-\text{Sat}})$ (resp. $\Pi^{\mathcal{M}}(S_{\text{BBU}_2-\text{Sat}})$).

In order to evaluate the steady-state call blocking probability in the system, in BBU₁, and in BBU₂ by considering different traffic loads, we vary the arrival rate λ of calls from 110 to 180. It can be observed through Fig. 5 that the increase of traffic load does not influence the call blocking probabilities until $\lambda = 160$, which is due to the resource availability. Nevertheless, as the offered traffic increases, the blocking probabilities also increase, but it remains acceptable despite using two BBUs and given the traffic load that reaches 180 calls per minute. Therefore, we can conclude that VMHAS has a higher acceptance

rate of connections request; hence more UEs can be served. As observed, also, the call blocking probability of BBU₂ is slightly lower than the call blocking probability of BBU₁. This slight difference between these two BBUs is because when both BBUs have the same current traffic load, the incoming call will be assigned to BBU₁.

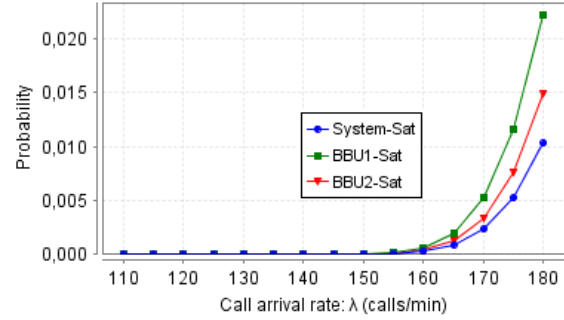


Figure 5: Steady-state call blocking probabilities.

5.1.2 $\mathcal{E}_{=?}(\text{true})$

We use CSRL (Haverkort et al., 2002) logic to express requirements related to the occupation rate for each BBU, and the switching between levels. Hence, we enrich PRISM \mathcal{M} model with the reward functions. Recall that the verification of this formula is given by Eq. 3.

- **Mean Occupation Rate:** In order to check that VMHAS ensures the load balancing between BBUs which is important in the system, we evaluate the mean occupation rate for BBU₁ (resp. BBU₂) by enriching our PRISM model with the reward function ρ_{BBU_1} (resp. ρ_{BBU_2}). We assign to each state $s = (i, j, l)$ the reward values $\rho_{\text{BBU}_1}(s) = 100(i)/Vmax$ and, $\rho_{\text{BBU}_2}(s) = 100(j)/Vmax$.

As observed in Fig. 6, when the traffic load increases, the mean occupation rate for each BBU increases. This is trivial because when the number of calls increases, the number of occupied VMs increases too. In addition, it can be observed that the mean occupation rate attains a significant percentage (nearly 90%) because the arrival rate per minute is significant (180 calls per minute) relative to the number of VMs in two BBUs. It is also remarkable that the difference between the two curves is very slight because when the call arrives, it will be assigned to the available VM in the least load BBU. By default, when the current traffic load is the same in the two BBUs, the first BBU will serve the incoming call. All that shows that our VMHAS ensures the load balancing between BBUs.

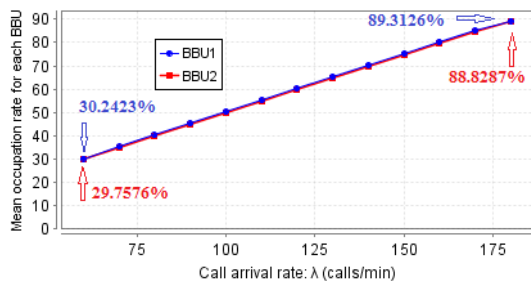


Figure 6: Steady-state occupation rate for each BBU.

- **Switching between Levels:** In order to evaluate the switching degree between levels depending on the hysteresis deactivating thresholds and the traffic load, we enrich the PRISM \mathcal{M} model with the reward function ρ_{level} . We assign to each state $s = (i, j, l)$ the reward value:

$$\rho_{level}(s) = l. \tag{9}$$

We present in the steady-state the switching between the first and the second level while changing T_1 (see Fig. 7), and the switching between the second and the third level while changing T_2 (see Fig. 8). Recall that according to Eq. 7 and Eq. 8, the size of the obtained \mathcal{M} (states and transitions number) will change with the variation of T_1 or T_2 . In fact, by decreasing T_1 , the size of \mathcal{M} increases which is confirmed by the experiment results obtained in Table 2, and that is due to the increase of state number in S_2 . Similarly, the size of \mathcal{M} increases too for decreasing values of T_2 (see Table 3) because the number of states increases in S_3 .

Table 2: The size of the model by varying T_1 .

T_1	States	Transitions
6	12329	36706
14	12169	36242
22	11881	35394
30	11465	34162
38	10921	32546

In Fig. 7, we can observe that by decreasing T_1 , the VMs of the second level stay active despite the decrease in traffic load, which is observable in the highest curve that still plates when λ is decreased from 100 to 25. Whereas, in the lowest curve, the second level is deactivated rapidly because the value of T_1 is near to V_{m1} .

Now, in Fig. 8, we illustrate the switching from the third to the second level by decreasing the traffic load. Similarly, the transition from the third to the second level is done quickly by values of T_2 near to V_{m2} . Whereas, by decreasing values of

Table 3: The size of the model by varying T_2 .

T_2	States	Transitions
42	12441	37074
46	12089	36026
50	11705	34882
54	11289	33642
58	10841	32306

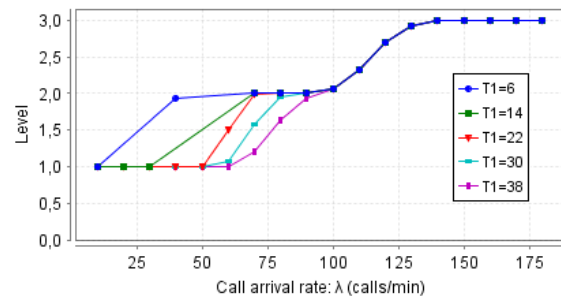


Figure 7: Steady-state switching between levels while changing T_1 .

T_2 , the deactivation of level three will make more time despite the decrease in traffic load.

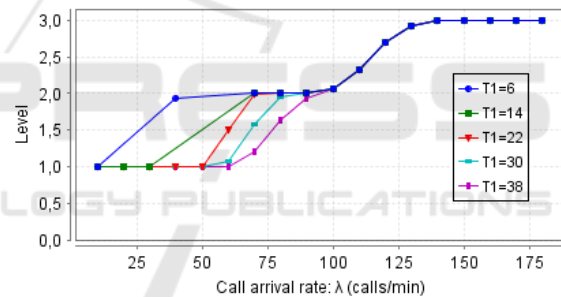


Figure 8: Steady-state switching between levels while changing T_2 .

5.2 Checking Transient-state Formulas

The verification of transient-state formulas at time t needs the computation of transient-state distribution $\Pi_s^{\mathcal{M}}(t)$, which depends on the initial state s , of the considered \mathcal{M} .

5.2.1 $\mathcal{P}_{=?}(true \mathcal{U}^{[t,t]} \phi)$

The verification of this formula is presented by Eq. 1. We will compute probabilities for VMHAS considering the initial state $s = (0, 0, 1)$. We suppose that at $t = 0$ all VMs are empty.

- $\mathcal{P}_{=?}(true \mathcal{U}^{[t,t]} \text{System.Sat})$: The verification of this formula is performed by the computation of the transient blocking probability of calls in two

BBUs at time t in the considered \mathcal{M} . This measure is equal to $\Pi_s^{\mathcal{M}}(S_{System_Sat}, t)$.

- $\mathcal{P}_{= ?}(true \mathcal{U}^{[t,t]} BBU_1_Sat)$: The verification of this formula is performed by the computation of the transient blocking probability of calls in BBU₁ at time t in the considered \mathcal{M} . This measure is equal to $\Pi_s^{\mathcal{M}}(S_{BBU_1_Sat}, t)$.
- $\mathcal{P}_{= ?}(true \mathcal{U}^{[t,t]} BBU_2_Sat)$: The verification of this formula is performed by the computation of the transient blocking probability of calls in BBU₂ at time t in the considered \mathcal{M} . This measure is equal to $\Pi_s^{\mathcal{M}}(S_{BBU_2_Sat}, t)$.

In order to evaluate the transient-state call blocking probability in the system and in each BBU, we fix $\lambda = 160$. It is observable through Fig. 9 that despite the heavy traffic load, the values of blocking probabilities (in the system, in BBU₁ and in BBU₂) are small. We note that the two BBUs saturation curves have similar probabilities. Note that the difference between the two curves is explained by when the two BBUs have the same number of occupied VMs, the incoming call will be assigned to the first BBU.

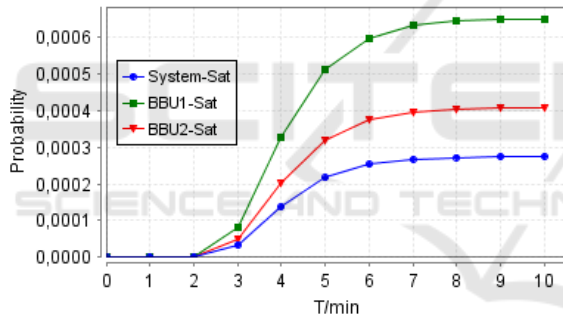


Figure 9: Transient-state call blocking probabilities.

5.2.2 $\mathcal{E}_{= ?}^t(true)$

We use the reward value presented in Eq. 9 in the transient case to evaluate the activation level at time t , depending on traffic load. this evaluation is illustrated in Fig. 10.

As observed, when the call arrival rate is light ($\lambda = 60$ calls/min), only the first level of VMs in two BBUs is activated to accept calls. However, the remaining two levels are deactivated because they are not needed. Nevertheless, when the traffic load increases ($\lambda = 90$ calls/min) and the number of occupied VMs attains the first hysteresis activating threshold, the second level of VMs will be activated in the two BBUs. When the traffic load is very high ($\lambda = 160$ calls/min), the third level of VMs will be activated.

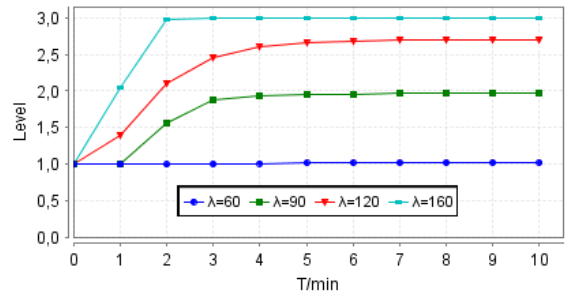


Figure 10: Transient-state activated level for BBUs.

6 CONCLUSION

In this paper, we have presented formal modelling and verification of call admission control strategy in the context of C-RAN. We have proposed a new CAC scheme, called Virtual Machine Hysteresis Allocation Strategy (VMHAS), based on two activation (resp. deactivation) hysteresis thresholds. We have developed the algorithmic description and the Markovian model of the proposed VMHAS. Then, we have used CSL logic to express the performance requirements of calls in terms of diminishing blocking probability, ensuring the load balancing between BBUs, and adjusting the number of active VMs being provisioned for the current traffic load. The performance analysis is performed using PRISM by checking CSL formulas in the transient and the steady-state of the system. Results show that the proposed model could have an acceptable blocking probability considering a high call arrival rate. Furthermore, it allowed load balancing between the BBUs and an active number of VMs according to the current traffic.

In the future, we will extend this work by performing additional performance measures to evaluate energy consumption and taking into account the eventual failure in the system by presenting a performativity model.

REFERENCES

- Al-Maitah, M., Semenova, O. O., Semenov, A. O., Kulakov, P. I., and Kucheruk, V. Y. (2018). A hybrid approach to call admission control in 5G networks. *Advances in Fuzzy Systems*, 2018.
- Aziz, A., Sanwal, K., Singhal, V., and Brayton, R. (2000). Model-checking continuous-time Markov chains. *ACM Transactions on Computational Logic (TOCL)*, 1(1):162–170.
- Bhamare, D., Erbad, A., Jain, R., Zolanvari, M., and Samaka, M. (2018). Efficient virtual network function

- placement strategies for cloud radio access networks. *Computer Communications*, 127:50–60.
- Checko, A., Christiansen, H. L., Yan, Y., Scolari, L., Kardaras, G., Berger, M. S., and Dittmann, L. (2014). Cloud RAN for mobile networks—a technology overview. *IEEE Communications surveys & tutorials*, 17(1):405–426.
- Chen, K. and Duan, R. (2011). C-RAN the road towards green RAN. *China Mobile Research Institute, white paper*, 2.
- Clarke, E. M., Emerson, E. A., and Sistla, A. P. (1986). Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 8(2):244–263.
- Duan, L., Zhan, D., and Hohnerlein, J. (2015). Optimizing cloud data center energy efficiency via dynamic prediction of CPU idle intervals. In *2015 IEEE 8th International Conference on Cloud Computing*, pages 985–988. IEEE.
- Gakhar, K., Achir, M., and Gravey, A. (2006). Dynamic resource reservation in IEEE 802.16 broadband wireless networks. In *2006 14th IEEE International Workshop on Quality of Service*, pages 140–148. IEEE.
- Haberland, B., Derakhshan, F., Grob-Lipski, H., Klotsche, R., Rehm, W., Schefczik, P., and Soellner, M. (2013). Radio base stations in the cloud. *Bell Labs Technical Journal*, 18(1):129–152.
- Halberstadt, S., Kofman, D., and Gravey, A. (1995). A congestion control mechanism for connectionless services offered by ATM networks. In *International Workshop on Performance Modelling and Evaluation of ATM Networks*, pages 57–73. Springer.
- Haverkort, B., Cloth, L., Hermanns, H., Katoen, J.-P., and Baier, C. (2002). Model checking performability properties. In *Proceedings International Conference on Dependable Systems and Networks*, pages 103–112. IEEE.
- Khan, M., Alhumaima, R., and Al-Raweshidy, H. (2015). Quality of service aware dynamic BBU-RRH mapping in cloud radio access network. In *2015 International Conference on Emerging Technologies (ICET)*, pages 1–5. IEEE.
- Kulkarni, V. G. (2016). *Modeling and analysis of stochastic systems*. Chapman and Hall/CRC.
- Kwiatkowska, M., Norman, G., and Parker, D. (2005). Probabilistic model checking in practice: Case studies with PRISM. *ACM SIGMETRICS Performance Evaluation Review*, 32(4):16–21.
- Kwiatkowska, M., Norman, G., and Parker, D. (2011). PRISM 4.0: Verification of probabilistic real-time systems. In *International conference on computer aided verification*, pages 585–591. Springer.
- Levy, H., Mendelson, T., and Goren, G. (2004). Dynamic allocation of resources to virtual path agents. *IEEE/ACM Transactions on networking*, 12(4):746–758.
- Lin, Y., Shao, L., Zhu, Z., Wang, Q., and Sabhikhi, R. K. (2010). Wireless network cloud: Architecture and system requirements. *IBM Journal of Research and Development*, 54(1):4–1.
- Sigwele, T., Alam, A. S., Pillai, P., and Hu, Y. F. (2017a). Energy-efficient cloud radio access networks by cloud based workload consolidation for 5G. *Journal of Network and Computer Applications*, 78:1–8.
- Sigwele, T., Pillai, P., Alam, A. S., and Hu, Y. F. (2017b). Fuzzy logic-based call admission control in 5G cloud radio access networks with preemption. *EURASIP Journal on Wireless Communications and Networking*, 2017(1):1–10.
- Sigwele, T., Pillai, P., and Hu, Y. F. (2014). Call admission control in cloud radio access networks. In *2014 International Conference on Future Internet of Things and Cloud*, pages 31–36. IEEE.
- Sigwele, T., Pillai, P., and Hu, Y. F. (2015). Elastic call admission control using fuzzy logic in virtualized cloud radio base stations. In *International Conference on Wireless and Satellite Systems*, pages 359–372. Springer.
- Urgaonkar, R., Kozat, U. C., Igarashi, K., and Neely, M. J. (2010). Dynamic resource allocation and power management in virtualized data centers. In *2010 IEEE Network Operations and Management Symposium-NOMS 2010*, pages 479–486. IEEE.
- Wang, K., Zhou, W., and Mao, S. (2017). On joint BBU/RRH resource allocation in heterogeneous cloud-RANs. *IEEE Internet of Things Journal*, 4(3):749–759.
- Younes, S. and Benmbarek, M. (2017). Performance analysis of multi-services call admission control in cellular network using probabilistic model checking. In *International Conference on Verification and Evaluation of Computer and Communication Systems*, pages 17–32. Springer.
- Younes, S. and Idi, M. (2018). Steady-state performability analysis of call admission control in cellular mobile networks. In *International Conference on Model and Data Engineering*, pages 5–16. Springer.